

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2010

Bc. Jan Bešina



VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Katedra měřicí a řídicí techniky



**SOFTWAREVÝ SIMULÁTOR SADY MĚŘICÍCH
PŘÍSTROJŮ PROGRAMOVATELNÝCH DLE
STANDARDU SCPI S PODPOROU DÁLKOVÉHO
OVLÁDÁNÍ PŘÍSTROJOVÝM OVLADAČEM
STANDARDU VISA PLUG AND PLAY**

**SET OF SIMULATED PROGRAMMABLE
INSTRUMENTS REMOTELY CONTROLLED BY SCPI
COMMANDS BY USING VISA PLUG AND PLAY
INSTRUMENT DRIVERS**

Zadání diplomové práce

Student: **Bc. Jan Bešina**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2601T004 Měřicí a řídicí technika
Téma: Softwarový simulátor sady měřicích přístrojů programovatelných dle standardu SCPI s podporou dálkového ovládání přístrojovým ovladačem standardu VISA Plug and Play

Set of Simulated Programmable Instruments Remotely Controlled by SCPI Commands by using VISA Plug and Play Instrument Driver

Zásady pro vypracování:

1. Seznámení se s problematikou programování měřicích přístrojů dle standardu SCPI.
2. Seznámení se s problematikou ovládání měřicích přístrojů přístrojovými ovladači standardu VISA Plug and Play.
3. Definice typů a vlastností simulovaných přístrojů.
4. Návrh a vývoj SW aplikace simulátoru přístrojů, vývoj přístrojového ovladače.
5. Ověření funkčnosti aplikace a zhodnocení použité technologie.

Seznam doporučené odborné literatury:

1. *LabVIEW Fundamentals* [online]. Austin: National Instruments, 08/2005 [cit.2009-09-09]. Dostupné z: < <http://www.ni.com/pdf/manuals/374029a.pdf> >. 165s.
2. *LabVIEW FPGA module User Manual* [online]. Austin: National Instruments, 03/2004 [cit.2009-09-09]. Dostupné z: < <http://www.ni.com/pdf/manuals/374690b.pdf> >. 62s.
3. ŽÍDEK, J. *Virtuální instrumentace na bázi grafického programování*. Ostrava: VŠB-TU Ostrava, 2003. 125s. ISBN 80-248-0298-8.
4. HAVLÍČEK, J.; VLACH, J.; VLACH, M.; VLACHOVÁ, V. *Začínáme s LabVIEW*. Praha: BEN, 2008. 248 s. ISBN 978-80-7300-245-9

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení:

„Místopřísežně prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Poděkování:

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce panu doc. Ing. Petru Bilíkovi Ph.D. , za rady, konzultace a připomínky, které mi byly nápomocné při tvorbě této práce.

V Orlové dne: 7.5. 2010

Jan Bešina

Abstrakt:

Předkládaná diplomová práce se věnuje problematice vzdálené komunikace s měřicími přístroji s využitím virtuální instrumentace. Cílem této práce je vytvoření softwarové náhrady několika měřících přístrojů pro účely výuky vzdáleného řízení měřících přístrojů.

Práce je rozdělená do několika částí. Po úvodní teoretické části následuje část praktická, která se zabývá vytipováním vhodných přístrojů a definováním jejich vlastností. Dalším krokem praktické části práce je vytvoření samotných přístrojů a množin příkazů, jimiž se půjdou tyto přístroje ovládat. V závěru praktické části této práce je vytvoření přístrojových ovladačů pro zjednodušení vytváření aplikací pro vzdálené ovládání simulovaných přístrojů.

V závěru práce je výsledná aplikace, včetně přístrojových ovladačů, otestována a je zhodnoceno použité řešení.

Klíčová slova: LabVIEW, SCPI, VISA, VXI Plug & Play přístrojový ovladač, vzdálené řízení

Abstract:

The presented diploma thesis deals with virtual instrumentation and remote control of measurement instruments. The main purpose of this work is choosing the appropriate instruments for designing and creating a simulation in the instruments themselves and also the communication with them.

The thesis is divided into several parts. The introductory theoretical part is followed by a practical part that intends to choose the appropriate instruments and define their character. Furthermore, the work deals with creating the instruments themselves and their commands by which the instruments will be operated. The final part of the practical section is intended to create instruments' drivers that serve as a simplification of formation of the applications for long-term operation of simulated instruments.

The resultant application is, along with instrument's drivers, in the final part tested and the used solutions are evaluated.

Keywords: LabVIEW, SCPI, VISA, VXI Plug & Play Instrument Driver, Remote Control

Seznam použitých symbolů a značek:

Symbol	Název
Hz	Jednotka frekvence, Hertz
V	Jednotka napětí, Volt
ms	Jednotka času, milisekunda
subVI	Podprogram vytvořený ve vývojovém prostředí LabVIEW, který je volán z jiného VI
VI	Program vytvořený ve vývojovém prostředí LabView

Seznam tabulek:

Tab. 1	Signály používané pro rozhraní GPIB	- 5 -
Tab. 2	Srovnání požadavků na LAN a LXI přístroje	- 7 -
Tab. 3	Povinné příkazy a dotazy obecné úrovně podle IEE 488.2	- 8 -
Tab. 4	Popis jednotlivých bitů STB registru.....	- 14 -
Tab. 5	Popis jednotlivých bitů Operation status registru	- 15 -
Tab. 6	Popis jednotlivých bitů Standard Event registru.	- 15 -
Tab. 7	Popis jednotlivých bitů Questionable status registru.	- 16 -
Tab. 8	Význam čísel chyb použitých v simulátoru.	- 16 -
Tab. 9	Napěťové rozsahy pro osciloskop.....	- 25 -
Tab. 10	Závislost počtu vzorků na zvolené časové konstantě	- 27 -
Tab. 11	Porty určené pro komunikaci s přístroji	- 30 -

Seznam obrázků:

Obr. 1	Vztah latence a přenosové rychlosti u jednotlivých rozhraní.....	- 6 -
Obr. 2	Návaznost standardů IEEE 488.1, IEEE 488.2 a SCPI.....	- 9 -
Obr. 3	Vnější struktura přístrojového ovladače	- 10 -
Obr. 4	Vnitřní struktura přístrojového ovladače	- 11 -
Obr. 5	Složení STB registru	- 13 -
Obr. 6	Čelní panel zdroje reálného napětí Agilent E3631A.....	- 18 -
Obr. 7	Vzhled čelního panelu simulátoru zdroje napětí	- 18 -
Obr. 8	Zdrojový kód programu simulující zdroj napětí	- 19 -
Obr. 9	Čelní panel reálného generátoru funkcí Agilent 33220A.....	- 20 -
Obr. 10	Čelní panel simulátoru generátoru funkcí	- 20 -
Obr. 11	Část programu pro vygenerování jednoho vzorku signálu.....	- 21 -
Obr. 12	Část programu pro přičtení šumu a DC offsetu k signálu	- 21 -
Obr. 13	Výsledný zdrojový kód generátorů funkcí	- 22 -
Obr. 14	Čelní panel reálného digitálního multimetru Agilent 34410A	- 22 -
Obr. 15	Čelní panel simulátoru digitálního multimetru	- 23 -
Obr. 16	Výsledný zdrojový kód digitálního multimetru	- 24 -
Obr. 17	Část programu, obsluhující měření zvolené hodnoty	- 24 -
Obr. 18	Čelní panel reálného osciloskopu Agilent MSO6012A	- 26 -
Obr. 19	Čelní panel simulátoru osciloskopu	- 26 -
Obr. 20	Část kódu osciloskopu pro nalezení bodu splnění triggrovací podmínky	- 27 -
Obr. 21	Blokové schéma komunikační smyčky na straně simulátoru	- 28 -
Obr. 22	Blokové schéma komunikační smyčky na řídicí aplikaci.....	- 28 -
Obr. 23	Část programu obsluhující komunikaci na straně simulátoru	- 28 -
Obr. 24	Vývojový diagram komunikace.....	- 29 -
Obr. 25	Část kódu obsluhující přijetí zprávy	- 30 -
Obr. 26	Část kódu obsluhující posílání zprávy	- 30 -
Obr. 27	Kód subVi pro rozpoznání příkazu	- 31 -
Obr. 28	Vývojový diagram rozpoznání příkazu.....	- 32 -
Obr. 29	Hlavní obrazovka simulátoru.....	- 33 -
Obr. 30	Část obrazovky obsahující nastavení přístrojů.....	- 33 -
Obr. 31	Část programu zobrazující událost „timeout“	- 35 -
Obr. 32	Obsluha změny časové konstanty	- 35 -
Obr. 33	Příklad souboru jednotlivých přístrojových ovladačů pro osciloskop	- 36 -
Obr. 34	Příklad přístrojového ovladače pro vyčtení dat změřených osciloskopem	- 36 -
Obr. 35	Odeslání příkazu.....	- 37 -
Obr. 36	Přijetí zprávy	- 37 -
Obr. 37	Vzhled čelního panelu programu pro vzdálené ovládání zdroje napětí.....	- 38 -
Obr. 38	Zdrojový kód programu pro vzdálené ovládání zdroje napětí.....	- 38 -
Obr. 39	Vzhled čelního panelu programu pro vzdálené ovládání generátoru	- 39 -
Obr. 40	Zdrojový kód programu pro vzdálené ovládání generátoru	- 39 -
Obr. 41	Čelní panel programu pro vzdálené ovládání digitálního multimetru	- 40 -
Obr. 42	Zdrojový kód programu pro vzdálené ovládání digitálního multimetru.....	- 40 -
Obr. 43	Vzhled čelního panelu programu pro vzdálené ovládání osciloskopu	- 41 -
Obr. 44	Zdrojový kód programu pro vzdálené ovládání osciloskopu	- 41 -

OBSAH:

Úvod	- 1 -
1 Komunikace s přístroji	- 2 -
1.2 Komunikační rozhraní používané pro komunikaci s dálkově ovládanými přístroji ..	- 3 -
1.2.1 RS-232.....	- 3 -
1.2.1.1 Základní parametry rozhraní RS-232.....	- 3 -
1.2.2 RS-422.....	- 4 -
1.2.3 RS-485.....	- 4 -
1.2.4 GPIB.....	- 4 -
1.2.5 ETHERNET	- 5 -
1.2.6 LXI.....	- 6 -
1.2.7 USB.....	- 7 -
1.3 Standard IEEE 488.1	- 7 -
1.4 Standard IEEE 488.2	- 7 -
1.5 SCPI standard.....	- 8 -
1.6 Přístrojové ovladače standardu VXI Plug and Play	- 9 -
1.6.1 Model přístrojového ovladače	- 10 -
1.6.2 Vnější struktura přístrojového ovladače	- 10 -
1.6.3 Vnitřní struktura přístrojového ovladače	- 11 -
1.7 Registry stavů a událostí přístroje.....	- 13 -
2 Popis jednotlivých přístrojů.....	- 17 -
2.1 Simulátor zdroje napětí.....	- 17 -
2.1.1 Čelní panel simulátoru zdroje napětí	- 18 -
2.1.2 Kód simulátoru zdroje napětí.....	- 19 -
2.2 Simulátor generátoru funkcí	- 19 -
2.2.1 Čelní panel generátoru funkcí.....	- 20 -
2.1.2 Kód simulátoru generátoru funkcí	- 21 -
2.3 Simulátor digitálního multimetru.....	- 22 -
2.3.1 Čelní panel simulátoru digitálního multimetru	- 23 -
2.1.2 Kód simulátoru digitálního multimetru.....	- 24 -
2.4 Simulátor osciloskopu	- 25 -
2.4.1 Čelní panel simulátoru osciloskopu	- 26 -
2.1.2 Kód simulátoru osciloskopu	- 27 -
3 Komunikace simulovaný přístroj – uživatel	- 28 -

4 Rozpoznání příkazů pro dálkové ovládání přístrojů	- 31 -
5 Aplikace simulátoru čtyř přístrojů	- 33 -
6 Přístrojové ovladače pro simulované přístroje.....	- 36 -
6.1 Příklad vzdáleného ovládání pro zdroj napětí	- 38 -
6.2 Příklad vzdáleného ovládání pro generátor funkcí	- 39 -
6.3 Příklad vzdáleného ovládání pro digitální multimetr	- 40 -
6.4 Příklad vzdáleného ovládání pro osciloskop.....	- 41 -
7 Testování a ověření funkčnosti aplikace.....	- 42 -
Závěr	- 43 -
Použité zdroje a literatura.....	- 45 -
Seznam příloh	- 46 -

Úvod

Tématem této diplomové práce je vytvoření softwarové náhrady neboli simulátoru několika základních přístrojů využívaných v měřicí technice. Důvodem pro vývoj simulátoru měřicích přístrojů je potřeba nástroje, který umožní studentům předmětů souvisejících s automatizací měření a testováním, praktické použití přístrojových ovladačů pro komunikaci s měřicími přístroji. Vybavit každé z 16 pracovišť učebny programovatelnými zdroji, generátory, multimetry a osciloskopy je finančně velice náročné. Vyvinutý simulátor nahradí potřebné přístrojové vybavení s ohledem na možnost výuky komunikace s přístroji. Výsledek celé této diplomové práce bude následně sloužit pro potřeby výuky komunikace s měřicími přístroji.

Tato práce je rozdělena do dvou částí. První část předkládané práce se zabývá teoretickými poznatky. Na teoretickou část plynule navazuje část praktická, která řeší stěžejní úkol diplomové práce.

V teoretické části jsou popsány výhody a možnosti vzdálené komunikace s přístroji. Také je zde rozebráno, jaké jsou možná rozhraní, které lze pro komunikaci s přístroji použít. Nedílnou součástí teoretické části této práce je popsání standardů pro tvorbu příkazů a standardů pro vytváření přístrojových ovladačů.

Další, stěžejní částí je vlastní tvorba sady přístrojů, která začíná výběrem jednotlivých typů přístrojů a definováním jejich vlastností. Po samotném definování typů přístrojů a jejich vlastností je důležitou částí vytvoření souboru příkazů pro ovládání těchto přístrojů. Následně se může přejít k tvorbě jednotlivých částí programu. V této práci je popsáno vytvoření části obsluhující komunikaci simulátoru s aplikacemi pro ovládání pomocí příkazů, části programu pro rozpoznání příkazů a také části pro simulaci funkcí jednotlivých vytipovaných přístrojů. Následně je popsán postup při tvorbě přístrojových ovladačů a vzorových příkladů pro aplikaci těchto ovladačů.

V závěru celé práce je otestována funkčnost celého softwaru včetně otestování vytvořených přístrojových ovladačů a vytvořených příkladů pro jejich aplikaci. Dále je zhodnocena použitá technologie.

1 Komunikace s přístroji

Komunikace se přístroji slouží hlavně ke dvěma účelům. Prvním důležitým účelem, kde můžeme komunikaci s přístroji použít, je vzdálená konfigurace přístroje, druhým neméně důležitým účelem je samotné vyčítání naměřených dat.

V první fázi se přístroj nakonfiguruje tak, aby jeho konfigurace byla co nejvhodnější k danému typu měření, které od něj uživatel požaduje. Tato konfigurace může proběhnout přímo na přístroji, pomocí prvků na čelním panelu určených ke konfiguraci, nebo pomocí příkazů pro vzdálené ovládání. V této fázi převládá tok informací od uživatele k přístroji.

V druhé fázi, kdy dochází k samotnému vyčítání měřených hodnot, při komunikaci převládá tok informací od přístroje k uživateli. Na čelním panelu přístroje jsou prvky určené pro předání informací uživateli. Tyto prvky jsou obvykle v podobě číselných či grafických displejů.

Vzhledem k možnosti propojení PC a samotných přístrojů, můžeme nahradit lidskou obsluhu počítačem, čímž získáme následující výhody:

- možnost automatizace celého procesu měření, a tím i zefektivnění časového hlediska náročnosti celého procesu
- vyloučení lidského faktoru z měřicího řetězce, což vede k eliminaci jednoho zdroje chyb
- možnost dalšího zpracování dat, která jsou již přenesena do PC
- možnost snadného archivování či sdílení naměřených dat
- možnost efektivního prezentování naměřených dat
- možnost jednoduchého uložení naměřených dat do souborů či databází a jejich jednoduché vyhledání

[1]

1.2 Komunikační rozhraní používané pro komunikaci s dálkově ovládanými přístroji

1.2.1 RS-232

Jedná o nejstarší komunikační rozhraní mezi měřicími přístroji a počítačem. Používalo se převážně k připojení nejrůznějších periférií jako například tiskáren, modemů, myší a podobně. V dnešní době se již tak hojně toto rozhraní nevyužívá.

Samotná komunikace pomocí RS-232 je poměrně jednoduchá. V jednom okamžiku prochází rozhraním pouze jeden bit posílaného bytu. Byte je rozdělen do časové posloupnosti jednotlivých bitů a ty jsou doplněny pomocnými bity pro řízení asynchronního přenosu. Bity, kterými je posílaná zpráva doplněna jsou start bit, paritní bit a stop bit.

Velkou výhodou tohoto rozhraní je jednoduchá kabeláž. Nejmenší počet vodičů pro obousměrný přenos je tři. Jsou to tyto vodiče: zem, vysílací a přijímací vodič. Komunikace může probíhat v obou směrech zároveň. Definice logických stavů je následující: -3 až -15 voltů pro stav logické jedničky a +3 až +15 voltů pro stav logické nuly.

Zpráva posílaná přes toto rozhraní je ve tvaru textové zprávy, kde jsou jednotlivé znaky zakódovány pomocí ASCII tabulky. Jak vysílač, tak přijímač musí být nastaveny na stejnou hodnotu základních parametrů přenosu, jinak přenos zprávy nebude úspěšný.

[1]

1.2.1.1 Základní parametry rozhraní RS-232

Modulační rychlost (baud rate) – je uváděna v baudech a při modulaci odpovídající dvěma stavům odpovídá přenosové rychlosti v bitech/s a nastavuje se v řadě hodnot 300, 600, 1200, 2400, 4800, 9600, 19200,

Počet datových bitů (data bits) – jednotkou přenosu je jeden byte, ale sériový přenos umožňuje počet datových bitů menší než 8 bitů –7 (stačí pro standardní tabulku ASCII obsahující kódy v rozmezí 0 až 127) nebo 8 (pro rozšířenou ASCII tabulku obsahující kódy v rozmezí 0 až 255) bitů.

Počet stop bitů (stop bits) – stop bit ukončuje při asynchronním přenosu jeden paket (sestavá se ze start bitu, datových bitů a, paritního bitu a stop bitů). Standardními hodnotami jsou 1, 1a půl nebo 2 bity. Stop bit slouží ke vzájemné synchronizaci hodin přijímače a vysílače, které jsou na sobě nezávislé a mohou se v rámci jednoho paketu rozejít. Zvýšením počtu stop bitů zvýšíme toleranci hodinové frekvence, ale také zvýšíme redundanci a zpomalíme přenos.

Parita (parity) – parita zajišťuje chybu přenosu na nejnižší úrovni. Nastavovanou hodnotou může být „no parity“ – bez kontroly parity, „odd“ – lichá kontrola parity, „even“ – sudá kontrola parity, „marked“ – paritní bit je vždy logická jednička a nebo „spaced“ – paritní bit je vždy logická nula. Paritní bit doplňuje datové bity na nastavenou paritu, což umožňuje rozpoznat jednu chybu v přenosu a zažádat si opakování přenosu na vyšší úrovni.

Hlavní nevýhodou sériového rozhraní je jeho malá propustnost a omezení na dva komunikující protějšky. V dnešní době je výskyt RS232 u běžných osobních počítačů minimální a rozhraní RS-232 je téměř vytlačeno rozhraním USB. V oblasti měření a testování je však stále používáno.

[1]

1.2.2 RS-422

Jedná se o sériové rozhraní, které využívá diferenční symetrický elektrický potenciál oproti RS-232, které využívá nesymetrický signál vůči zemi. Pro každý směr přenosu se používá dvou vodičů, čímž je dosaženo větší odolnosti vůči rušení a zvětšení maximální vzdálenosti, na kterou lze přístroje propojit pomocí tohoto rozhraní. RS-422 umožňuje komunikaci až deseti přístrojů.

[1]

1.2.3 RS-485

RS-485 je zdokonalením RS-422, které zvyšuje možný počet komunikujících protějšků až na třicet dva. Toto rozhraní je ideálním řešením pro vytvoření komunikační sítě zařízení, která komunikují s jedním PC. Pro oba směry komunikace se používá stejný pár vodičů s přepínáním podle směru komunikace. Rozhraní RS-485 je vytvořeno tak, že pomocí něho lze komunikovat i s přístroji, které obsahují komunikační rozhraní RS-422.

[1]

1.2.4 GPIB

Pro komunikaci s měřicími přístroji produkující větší objemy dat je nevhodné použít rozhraní RS-232, protože popisované rozhraní má malou propustnost dat. Při 9600 bodech je propustnost kolem 1 kByte/s, což je například pro osciloskopy zcela nedostačující, jelikož pro zobrazení jedné křivky, která je rozložena na 4096 bodů s rozlišením 8 bitů, je potřeba přibližně 4 kByty dat. K tomu účelu se mnohem lépe hodí rozhraní GPIB, kdysi označované také jako HP-IB nebo IEEE 488. Jedná se o paralelní rozhraní s teoretickou propustností okolo 1Mbyte/s.

Vůdčím společností pro oblast GPIB je National Instrument, která vyvinula několik inovačních řešení, včetně kontroléru pro rozhraní GPIB na bázi integrovaného obvodu NI TNT ASICs, který představuje řešení GPIB komunikace v jednom pouzdru integrovaného obvodu. Tento obvod lze využít při konstrukci vlastního přístroje, který má mít komunikační rozhraní GPIB, a to s propustností až 1,5Mbytů/s při protokolu IEEE 488.1 nebo 8Mbytů/s při protokolu HS 488.

[1]

V následující tabulce (Tab. 1) jsou uvedeny signály, z kterých je vytvořena struktura GPIB rozhraní.

Označení signálu	Popis signálu
DIO1 až DIO8	Datové vodiče pro přenos jednotlivých bitů bytu
NRFD (Not Ready For Data)	Handshakový signál – indikuje připravenost zařízení pro příjem dat, 1=připraveno, rozhoduje logický součin od všech zařízení
NDAC (Not Data Accepted)	Handshakový signál – indikuje konec zpracování přijatého byte, 1=byte zpracován, rozhoduje logický součin od všech zařízení
DAV (Data Valid)	Handshakový signál – platná data na datových vodičích, 0=platná data
ATN (attention)	Řídicí signál – rozlišuje příkazy a data 1 = příkazy a 0 = data
IFC (interface clear)	Řídicí signál – reset obvodů rozhraní
REN (remote enable)	Řídicí signál – rozlišuje lokální a dálkové řízení
SQR (service request)	Řídicí signál – zařízení potřebuje obsluhu řidiče sběrnice
EOI (end of identify)	Řídicí signál – buď ukončení zprávy, nebo identifikace při paralelním dotazu
8 krát GND	Zamkní signál – 6 x v twisted páru se signály DAV, NRFD, NDAC, IFC, SRQ, ATN a 1 x signálová zem, 1 x stínění

Tab. 1 Signály používané pro rozhraní GPIB

[1]

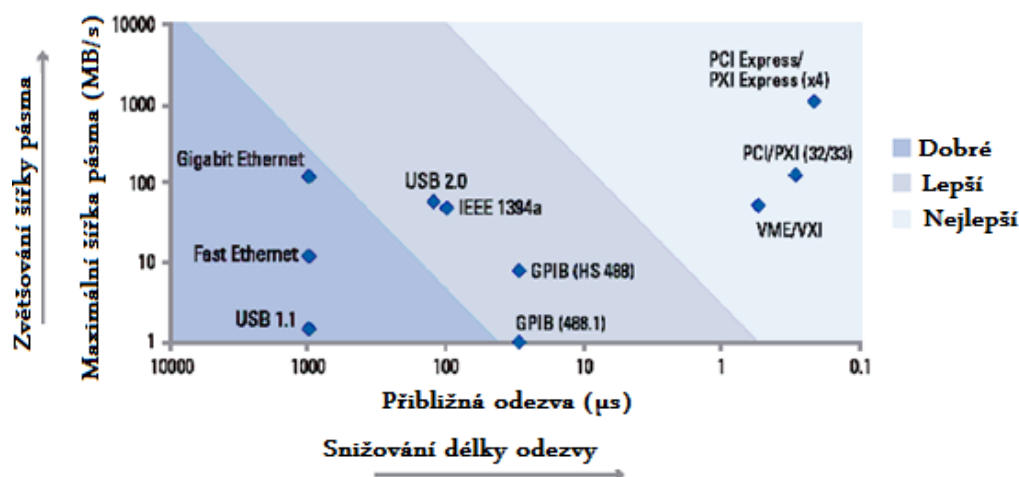
1.2.5 ETHERNET

Ethernetové rozhraní se většinou využívá pro vzájemné propojení počítačů v lokální síti. Přenosová rychlost postupně narůstá a všechny počítače bývají vybaveny tímto rozhraním, jež bylo hlavním důvodem rozšíření Ethernetu i do aplikační oblasti komunikace s měřicími přístroji.

Nevýhody Ethernet rozhraní:

- dlouhá latence s ohledem na časovou synchronizaci
- menší výběr přístrojů vybavených tímto rozhraním

Na následujícím obrázku (Obr. 1) je znázorněn vztah mezi dvěma podstatnými parametry pro komunikaci mezi přístroji. Těmito parametry jsou latence a přenosová rychlost. Z obrázku je patrné, že Ethernet rozhraní je z hlediska přenosové rychlosti srovnatelné s jinými rozhraními, ale je nesrovnatelně horší z hlediska latence. Délka latence je čas potřebný pro zahájení přenosu, proto hraje důležitou roli při degradaci přenosové schopnosti u přenosu velkého množství malých objemů dat.



Obr. 1 Vztah latence a přenosové rychlosti u jednotlivých rozhraní

[1]

1.2.6 LXI

Specifikace LXI je rozšíření rozhraní Ethernet v oblasti komunikace s přístroji. Přibyly zde tyto standardizované části:

- standardní podpora html stránky pro konfigurace přístroje – web server
- synchronizační a časovací mechanismy včetně IEEE 1588 – Precision Time Protocol
- hardwarový trigger externím signálem

Standard IEEE 1588 umožňuje synchronizaci prostřednictvím sítě LAN. Pomocí speciálního hardwaru lze dosáhnout časové synchronizace s přesností $\pm 100\text{ns}$. Tato schopnost je dostatečná pro synchronizaci zařízení s vzorkovací frekvencí pod 1MS/s na velkou vzdálenost.

Hardwarový trigger slouží ještě přesnější synchronizaci prostřednictvím sady LVDS (Low-Voltage Differential Signaling) na krátkou vzdálenost s využitím speciální kabeláže.

Většina přístrojů s LXI jsou pouze vylepšené verze stávajících přístrojů s rozhraním LAN. Pomocí přístrojů z této třídy je možné budovat rozlehlé měřicí systémy, kde jsou jednotlivé části od sebe vzdáleny na velké vzdálenosti. V následující tabulce (Tab. 2) je srovnání požadavků na přístroje vybavené rozhraním Ethernet (LAN) a přístroje vybavené modifikovaným rozhraním Ethernet (dle specifikace LXI).

Možnost	LAN přístroje	LXI přístroje
Lan rozhraní	vyžadováno	vyžadováno
Trigger vstup/výstup	jako možnost	jako možnost
Konfigurační panel přes web	jako možnost/typicky	vyžadováno
IVI – přístrojový ovladač	jako možnost/typicky	vyžadováno
Hardwarový trigger	jako možnost	jako možnost (vyžadováno pro třídu A)
IEE-1588	jako možnost	jako možnost (vyžadováno pro třídu A,B)

Tab. 2 Srovnání požadavků na LAN a LXI přístroje

[1]

1.2.7 USB

V dnešní době představuje USB velmi rozšířenou a levnou alternativu pro komunikaci s měřicími přístroji. Velkou výhodou tohoto rozhraní je velmi dobrá prostupnost a míra rozšíření na straně PC. Další velmi podstatnou výhodou tohoto rozhraní je to, že je možné přístroj připojit za provozu. K jednomu rozhraní je možno připojit i více přístrojů přes USB hub. USB automaticky detekuje připojený přístroj. Podporuje asynchronní i synchronní přenos dat. Při synchronním přenosu dat je garantován čas, který je potřeba k předání dat, ale negarantuje se bezchybnost samotného přenosu. Při asynchronním přenosu dat podporuje USB ošetření události, přičemž pro každou událost lze přiřadit prioritu, a tím zajistit její přednostní obsluhu.

USB ve verzi 2.0 umožňuje 40 krát větší přenosovou rychlost než verze 1.1 (480 Mbitů/s), ale také garantuje i řádové snížení doby latence.

Při použití konvertorů lze toto rozhraní použít i pro komunikaci s přístroji, které jsou vybavené koncepčně staršími komunikačními rozhraními, jako je například RS-232 nebo GPIB.

[1]

1.3 Standard IEEE 488.1

Tímto standardem jsou definovány standardní elektrické a mechanické vlastnosti pro konektory a kabely. Také tato norma definuje handshaking, adresování a obecný protokol pro přenos bytů z přístrojů do počítače. Tento standard neřeší význam jednotlivých přenesených bytů.

[2]

1.4 Standard IEEE 488.2

Tento standard navázal na předchozí specifikaci a je zpracován tak, aby zůstal kompatibilní s IEEE 488.1. Navrhuje způsob, jakým přístroje spolu komunikují, tak, aby to bylo zcela spolehlivé a vysoce účinné.

Do standardu IEEE 488.2 byly implementovány i standardní příkazy, které jsou v následující tabulce (Tab. 3). Tyto příkazy byly zárodkem jednotné syntaxe pro komunikaci s přístroji – SCPI standard.

Syntaxe příkazu	Skupina	Popis funkce příkazu
*IDN?	Systémová data	Identifikace přístroje
*RST	Vnitřní operace	Reset přístroje
*TST?	Vnitřní operace	Test přístroje
*OPC	Synchronizace	Oznámení dokončení příkazu
*OPC?	Synchronizace	Dotaz zda je dokončen příkaz
*WAI	Synchronizace	Vyčkání na dokončení příkazu
*CLS	Statusy a události	Vynulování statusů
*ESE	Statusy a události	Registr událostí
*ESE?	Statusy a události	Dotaz na registr událostí
*ESR?	Statusy a události	Dotaz na status registr
*SRE	Statusy a události	Registr požadavků
*SRE?	Statusy a události	Dotaz na registr požadavků
*STB?	Statusy a události	Vyčtení status byte

Tab. 3 Povinné příkazy a dotazy obecné úrovně podle IEE 488.2

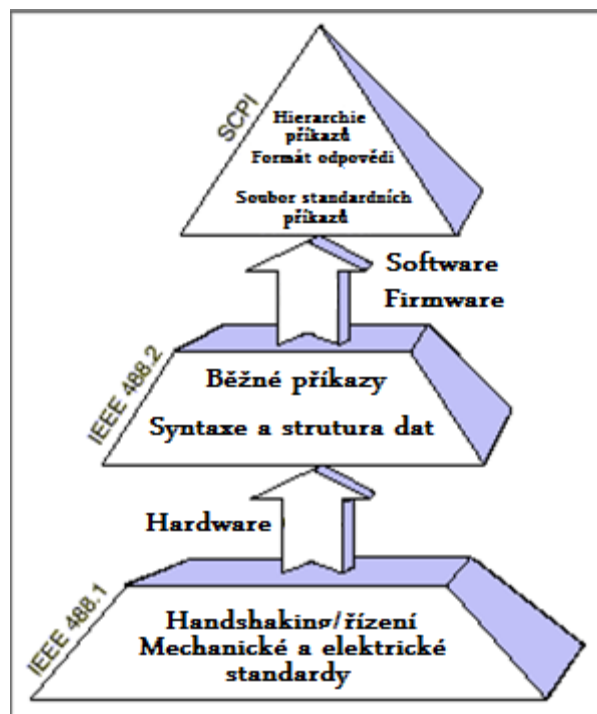
[2]

1.5 SCPI standard

Standard SCPI (Standard Commands for Programmable Instruments) je standard příkazů pro programovatelné přístroje. Je to jazyk založený na ASCII příkazech. SCPI příkazy jsou založeny na hierarchické struktuře, kde jsou příkazy děleny dle společného kořenu, čímž se vytvoří stromová struktura příkazu.

Standard SCPI rozšířil stávající standard IEEE 488.2 o definované komplexní příkazy pro všechny přístroje. Například díky SCPI může uživatel ovládat všechny SCPI kompatibilní voltmetry bez ohledu na výrobce nebo model přístroje, tyto přístroje budou vždy reagovat na stejný příkaz. Rovněž formát jejich odpovědí bude také vždy stejný.

SCPI standard v sobě zahrnuje standardy IEEE 488.1 a 488.2, což je znázorněno na následujícím obrázku (Obr. 2). Kombinací SCPI a IEEE 488.2 se dosáhne vyšší produktivity při tvorbě a použití přístrojových ovladačů. Uživatelé se mohou více soustředit na jiné problémy než je učení se jednotlivých příkazů a jejich syntaxí.



Obr. 2 Návaznost standardů IEEE 488.1, IEEE 488.2 a SCPI

SCPI standard definuje:

- Hierarchicky uspořádanou množinu SCPI příkazů pro ovládání měřících přístrojů
- Množinu obecných povinných příkazů – command commands – určených k identifikaci stavu a nastavení vnitřních registrů
- Formáty dat

[2]

1.6 Přístrojové ovladače standardu VXI Plug and Play

VXI Plug and Play přístrojový ovladač je soubor jednotlivých VI, které slouží k ovládání programovatelného přístroje. Každé VI odpovídá jednotlivým operacím, které lze s přístrojem provádět, jako například konfigurace, triggering a čtení změřených hodnot z přístroje. Přístrojový ovladač slouží uživatelům k usnadnění počáteční práce s programovatelným přístrojem a rovněž slouží k ušetření času při vývoji, a tím pádem vede i ke snížení nákladů výsledného programu. Velkou výhodou přístrojových ovladačů je to, že uživatel nemusí znát všechny programovací možnosti jednotlivých přístrojů. Pomocí otevřených a velmi dobře dokumentovaných přístrojových ovladačů, které si konečný uživatel může upravit dle libosti, může uživatel dosáhnout požadovaného výsledku. Vzhledem k modulárnímu designu, je úprava ovladačů velmi jednoduchá.

Výhody používání standardu architektury pro všechny VXI PnP přístrojové ovladače:

- Zlepšení konzistence přístrojových ovladačů pro koncové uživatele
- Zvýšení kvality ovladačů
- Minimalizování několikanásobného úsilí
- Zlepšuje snadnost použití pro koncové uživatele tím, že umožňuje stejné použití přístrojových ovladačů z různých zdrojů
- Zjednodušuje vývoj přístrojového ovladače ve prospěch tvůrců samotného přístrojového ovladače

Přístrojové ovladače obsahují vysokoúrovňové VI s intuitivními předními panely, tak aby koncoví uživatelé mohli rychle vyzkoušet a ověřit možnosti programovatelného přístroje. Koncoví uživatelé tímto nemusí znát nízkou úroveň ovládání samotného přístroje, jako jsou specifické přístroje a jejich syntaxi. Uživatelé vytvářejí aplikaci pro ovládání přístroje jednotlivých pomocí stavebních prvků vkládaných do jejich blokových schémat. Těmito stavebními prvky jsou přístrojové ovladače ve formě subVI.

[3]

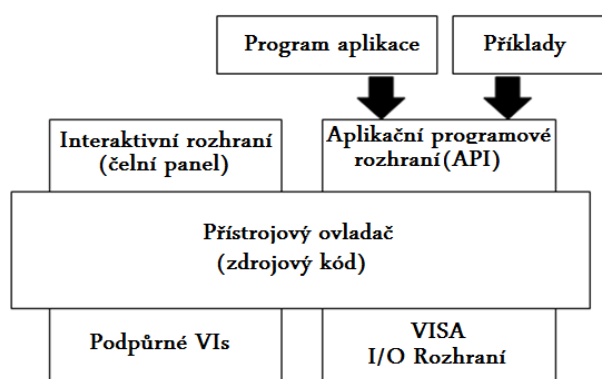
1.6.1 Model přístrojového ovladače

Mnoho programovatelných zařízení má velké množství funkcí a režimů. Díky této složitosti je nezbytné, poskytnout jednoduchý design modelu, který pomůže jak vývojářům přístrojových ovladačů, tak i samotným uživatelům, kteří pomocí těchto přístrojových ovladačů vyvíjejí aplikaci pro ovládání programovatelného zařízení. Model přístrojového ovladače je složen z vnější i vnitřní struktury. Vnější struktura ukazuje, rozhraní přístrojového ovladače s uživatelem a s ostatními softwarovými komponenty systému. Vnitřní struktura pak ukazuje vnitřní organizaci modulů přístrojového ovladače.

[3]

1.6.2 Vnější struktura přístrojového ovladače

Přístrojový ovladač je složen z VIs, kde uživatel volá na vyšší úrovni. Na následujícím obrázku jsou znázorněny interakce přístrojového ovladače se zbytkem systému.



Obr. 3 Vnější struktura přístrojového ovladače

Vnější struktura ukazuje interaktivní rozhraní i rozhraní pro programování aplikací. Rozhraní pro programování aplikací (API) je soubor přístrojových ovladačů ve formě jednotlivých VI, které může uživatel zavolat ve své aplikaci. Jako příklad může posloužit testovací aplikace, která pro svoji funkčnost používá přístrojové ovladače pro komunikaci s multimetrem nebo osciloskopem.

Uživatel se dozvídá všechny potřebné informace z čelního panelu přístrojového ovladače, ze kterých lze jednoduše pochopit, jaký vliv má ovladač na programovaný přístroj. Díky tomu mohou běžní uživatelé jednoduše tyto ovladače použít.

Softwarová architektura virtuálních nástrojů (VISA – Virtual Instrument Software Architecture) vstupně-výstupního rozhraní v LabView je množina funkcí, jenž slouží pro komunikaci s přístroji. VISA zakládá standart pro komunikaci na standardních komunikačních rozhraních (GPIB, USB, sériová). Tato architektura slouží k zefektivnění celého procesu vytváření aplikací pro měřicí přístroje.

Knihovna funkcí VISA jsou schopny realizovat předepsané výměny zpráv bez ohledu na fyzické rozhraní, které k této výměně bude využito. Typ rozhraní, které bude ke komunikaci využito, je parametrem funkce pro inicializaci komunikace, při které se vytváří proměnná ovladače. Prostřednictvím tohoto ovladače se provádí operace zápisu a čtení v rámci komunikace s přístrojem. Tento přístup umožňuje záměnu typu komunikačního rozhraní přístroje bez potřeby složitých změn v programu.

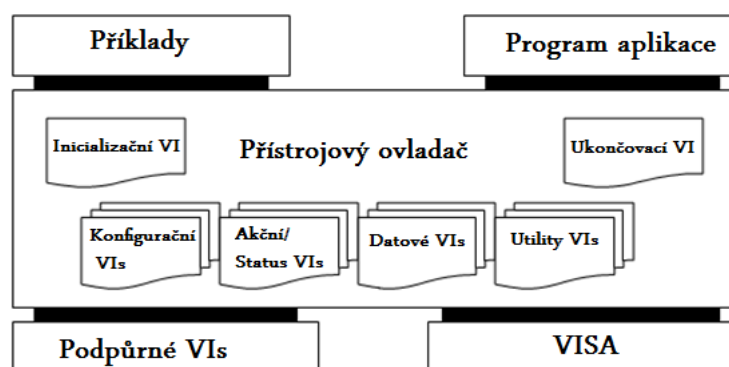
[3]

1.6.3 Vnitřní struktura přístrojového ovladače

Vnitřní struktura přístrojového ovladače definuje organizaci jednotlivých přístrojových ovladačů, a to tak, aby byly co nejvíce přehledně uspořádány. Pro největší přehlednost jsou přístrojové ovladače uspořádány dle jejich funkčnosti.

Na obrázku č. 4 je ukázaná vnitřní struktura přístrojových ovladačů. Jak je patrné z obrázku č. 4 mají koncoví uživatelé jednotlivé ovladače přehledně rozděleny a jednoduše k dispozici dle jejich potřeby.

Potom co se uživatel naučí pracovat s jedním přístrojovým ovladačem, lze většinu takto nabytých znalostí využít i na jiné přístrojové ovladače.



Obr. 4 Vnitřní struktura přístrojového ovladače

Rozdělení přístrojových ovladačů, lze provést do 6 skupin:

Inicializační VI - všechny soubory přístrojových ovladačů by měly obsahovat toto VI, jenž vytvoří spojení mezi softwarem a přístrojem. Je to první VI, které je při práci s programovatelným přístrojem použito. Optimálně umožňuje provést inicializaci zařízení.

Konfigurační VIs – je sbírkou jednotlivých VI, která umožňují nakonfigurování přístroje. Jejich počet závisí na počtu jedinečných schopností samotného přístroje. Po zavolání těchto VI, je zařízení připraveno k použití.

Akční/Status VIs – jsou to VI, která slouží k zahájení nebo ukončení zkušebního provozu měření. Tyto VI se liší od konfigurace, tím že nemění samotné nastavení přístroje. Samotný příkaz je závislý na aktuálním nastavení přístroje. Status VI, lze použít pro získání aktuálního stavu přístroje před nebo po dané operaci. Tyto VI se obvykle používají, vyžaduje-li další funkce tato data.

Datové VIs – tyto VI umožňují převedení dat z nebo do přístroje. Zahrnují například VI pro čtení měřené hodnoty nebo křivky či k nastavení digitálních výstupů nebo k přečtení digitálních vstupů.

Utility VIs – tyto VI vykonávají různé doplňující funkce, jako je například reset nebo self-test. Může také obsahovat funkce, které plní funkci kalibrace, uchování nastavení nebo obnovení nastavení z uchovaných hodnot.

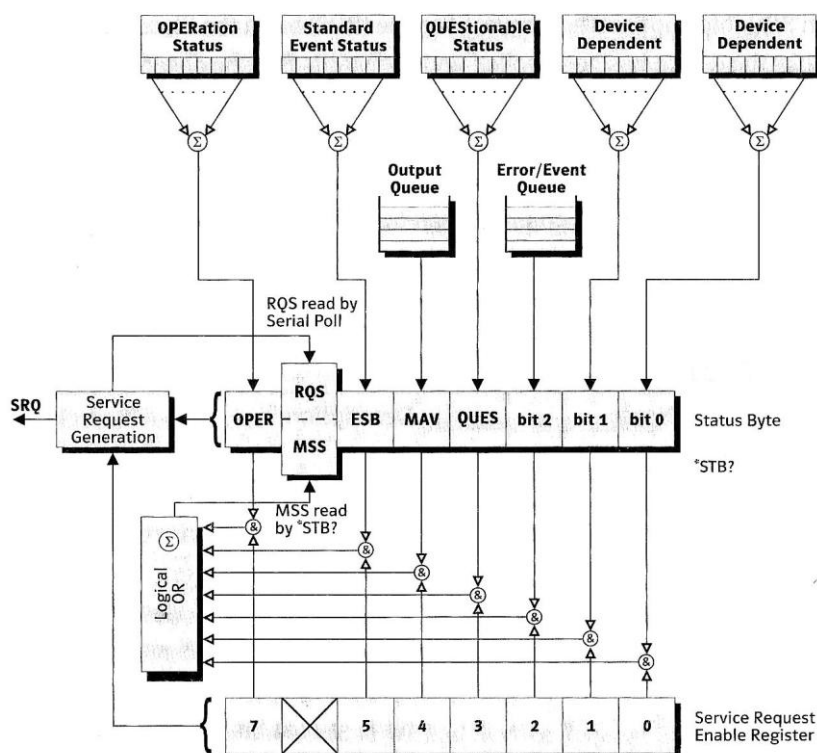
Ukončovací VI - toto VI ukončí spojení softwaru s přístrojem. Každý soubor přístrojových ovladačů by měl toto VI obsahovat.

Každá kategorie, s výjimkou inicializačních a ukončovačích obsahuje několik VI.

[3]

1.7 Registry stavů a událostí přístroje

Hlavním účelem registrů je podání informací o stavu zařízení a vyskytlých událostech. Nejdůležitějším registrem je registr označený jako STB (zkratka z anglického status byte register). Jeho grafická podoba je znázorněna na následujícím obrázku (Obr. 5).



Obr. 5 Složení STB registru

Popis jednotlivých bitů a vysvětlení jejich významů je v tabulce 4. Registr STB funguje jako souhrn informací ze všech registrů podávající informace o stavu zařízení. K obsluze STB registru slouží příkazy „*STB?“, „*SRE“ a „*SRE?“. Příkaz „*STB“ slouží k vyčtení aktuálního obsahu tohoto registru. Pomocí příkazu „*SRE“ se nastaví maska STB registru. Poslední příkaz „*SRE?“ slouží k zjištění aktuálního nastavení masky.

Bit	Označení	Jméno bitu	Popis
7	OPER	Operation Status	Obsahuje sumu jednotlivých bitů z OPERation status struktury.
6	RQS	Request Service	Indikuje vyžádání servisního zásahu. PO přečtení této hodnoty je bit vynulován. Logická „1“ se nastaví pouze v situaci, kdy vznikne nová událost vyžadující servisní zásah.
	MSS	Master Summary Status	Indikuje, že je status byte, který způsobil vyžádání servisního zásahu. Na tento bit má vliv nastavení masky pomocí příkazu *SRE.
5	ESB	Event Summary Bit	Obsahuje sumu jednotlivých bitů z registru standardní událostí.
4	MAV	Message Available	Indikuje, že výstupní zásobník obsahuje zprávu pro odeslání. Tento bit je nastaven do logické „1“ pakliže zásobník obsahuje aspoň jeden byte k odeslání. Logická nula je nastavená okamžitě po vyprázdnění tohoto zásobníku.
3	QUES	Questionable status	Obsahuje sumu jednotlivých bitů z QUEStionable status struktury.
2	Bit 2	Error/Event Queue	Signalizuje, že v zásobníku dotazů nebo chyb není prázdný. Tento bit je nastaven do logické „1“ pakliže zásobník obsahuje aspoň jeden byte k odeslání. Logická nula je nastavená okamžitě po vyprázdnění těchto zásobníků.
1	Bit 1	Device Dependent status	Nastavení toho bitu je závislé na specifikaci přístroje.
0	Bit 0	Device Dependent status	Nastavení toho bitu je závislé na specifikaci přístroje.

Tab. 4 Popis jednotlivých bitů STB registru

Operation status - tento registr je složen z 16 bitů vyjmenování a popis jednotlivých bitů je uveden níže. Typické přístroje nemusí mít využité všechny bity toho registru. Například přístroj sloužící jako zdroj signálu nebude mít využitý bit č. 4 signalizující, že zařízení právě teď měří. Nevyužité bity budou mít vždy hodnotu logické nuly. Význam jednotlivých bitů je popsán v tabulce 5. Mezi příkazy vztahující se k tomuto registru patří příkazy začínající „STATus:OPERation“. U toho registru lze také nastavit jeho maska určující, které bity chceme použít a které chceme zakázat. Logický součet všech bitů tohoto registru je vložena do sedmého bitu STB registru.

Bit	Označení bitu	Jméno bitu	Popis
15			Tento bit není využitý
14	PROG	Program running	Program, který je stažen z paměti, je v provozu.
13	INST	Instrument summary bit	Logický součet všech bitů.
8-12			Tyto bity jsou specifikovány zařízením.
7	CORR	Correcting	
6	ARM	Waiting for ARM	Zařízení vyčkává na ARM signál.
5	TRIG	Waiting for Trigger	Zařízení vyčkává na splnění trigrovací podmínky.
4	MEAS	Measuring	Signalizuje právě probíhající měření.
3	SWE	Sweeping	Signalizuje, že zařízení je v sweep režimu.
2	RANG	Ranging	Signalizuje, že zařízení právě mění jeho rozsah.
1	SETT	Setting	Signalizuje, že zařízení není připravené pro měření.
0	CAL	Calibration	Signalizuje právě probíhající kalibraci.

Tab. 5 Popis jednotlivých bitů Operation status registru

Standard Status – tento registr je součástí standardu IEEE 488.2. Struktura a popis jednotlivých bitů tohoto registru je v tabulce č. 6. Logický součet tohoto registru je vložen do pátého bitu STB registru. K tomu registru se vztahují příkazy „*ESR?“, který vypíše aktuální obsah tohoto registru. Příkaz „*ESE“ slouží k zjištění nastavené masky standard event registru a příkaz „*ESE?“ slouží k zjištění nastavené masky.

Bit	Označení	Jméno	Popis
7	PON	Power On	Zařízení je zapnuto.
6	URQ	User Request	Zařízení vyžaduje zásah uživatele. Například zmáčknutí tlačítka na čelním panelu.
5	CME	Command Error	Chyba příkazu.
4	EXE	Executive Error	Chyba přerušení.
3	DDE	Device Dependent Error	Chyba závislého zařízení.
2	QYE	Query Error	Chyba dotazu.
1	RQC	Request Control	Zařízení požaduje předat řízení komunikace.
0	OPC	Operation Complete	Signalizace ukončení všech operací.

Tab. 6 Popis jednotlivých bitů Standard Event registru.

Questionable status - tento registr je složen z 16-ti bitů. Popis jednotlivých bitů je uveden v tabulce 7. Tento registr podává informace o kvalitě jednotlivých aspektů zařízení. Report podány tímto registrem nemusí nutně znamenat fatální chybu. Signalizuje pouze možnost nenaplnění očekávání aplikace. Mezi příkazy vztahující se k tomuto registru patří příkazy začínající „STATUS:QUESTIONable“. U toho registru lze také nastavit jeho maska určující, které bity chceme použít a které chceme zakázat. Logický součet všech bitů tohoto registru je vložena do třetího bitu STB registru.

Bit	Označení	Jméno	Popis
15			Tento bit není využitý
14		Command warning	Varování o chybě, která není kritická.
13	INST	Instrument summary bit	Logický součet všech bitů.
12-9			Tyto bity jsou specifikovány zařízením.
8	CAL	Calibration	Provedená kalibrace nemá zaručenou správnost.
7	MOD	Modulation	Modulace nemá zaručenou správnost.
6	PHAS	Phase	Změřená fáze signálu je nejistá.
5	FREQ	Frequency	Změřená frekvence signálu je nejistá.
4	TEMP	Temperature	Změřená teplota je nejistá.
3	POW	Power	Změřený výkon signálu je nejistý.
2	TIME	Time	Změřená časová charakteristika nemá zaručenou správnost.
1	CURR	Current	Změřená proudová charakteristika je nejistý.
0	VOLT	Voltage	Změřená napěťová charakteristika je nejistý.

Tab. 7 Popis jednotlivých bitů Questionable status registru.

Output queue – (výstupní zásobník) tento zásobník slouží k dočasnému uložení odpovědi zařízení do doby než jej přečteno řídicí aplikací. V případě, že v tomto zásobníku se nachází nějaká hodnota je do STB registru na pozici čtvrtého bitu uložena logická jednička signalizující, která upozorňuje, že je připravená zpráva na odeslání. Po odeslání dat uložených v tomto zásobníku, jsou tyto data odtud vymazána.

Error/Event queue – (zásobník chyb/událostí) pokud není tento zásobník prázdný je na druhý bit STB registru uložena logická jednička, která na tuto situaci upozorňuje.

Příkazem „SYSTem“ERRor:NEXT?“, dojde k vyčtení chyby nebo události která se vyskytla jako první. Aby došlo k úplnému vyprázdnění toho zásobníku je tento příkaz potřeba zavolat tolikrát, než jako odpověď na něj bude vrácena hodnota 0, která značí, že v zásobníku není uložena žádná chyba. Na následující tabulce (Tab. 8) je uveden význam chyb, ke kterým může v dále navrhovaném simulátoru dojít.

Číslo chyby	Význam
0	Bez chyby
-100	Chyba syntaxe příkazu
-200	Chyba rozsahu

Tab. 8 Význam čísel chyb použitých v simulátoru.

[4]

2 Popis jednotlivých přístrojů

Jako přístroje vhodné k vytvoření simulátoru byly vybrány čtyři nejčastěji používané přístroje v měřicí technice: zdroj napětí, generátor funkcí, digitální multimetr a osciloskop. Definované vlastností jsou popsány v podkapitolách níže přímo u konkrétních přístrojů.

2.1 Simulátor zdroje napětí

Tento přístroj simuluje funkci zdroje napětí, u kterého je možno měnit velikosti výstupního napětí. U tohoto zdroje je možnost i použití napěťové rampy, u napěťové rampy je k dispozici nastavení počátečního napětí, koncového napětí, velikost jednotlivých kroků a času mezi jednotlivými kroky. Napětí zdroje se může zadat v mezích od 0V do +12.0V. Pro nastavení počátečního a koncového napětí u napěťové rampy lze nastavit jakékoliv napětí z rozsahu 0 V do +12.0 V, jedinou podmínkou je, aby tato dvě napětí nebyla shodná. Velikost kroku, tj. rozdíl napětí mezi jednotlivými kroky je možno zadat v rozmezí od 0.1V do 12V. V případě, že by nastala situace, že při dalším kroku bude překročeno koncové napětí, bude výstupní napětí zaokrouhleno na hodnotu zadaného koncového napětí nebo bude rovno maximálnímu (minimálnímu) napětí na výstupu zdroje. Těmito podmínkami je dosaženo toho, aby byl vždy při použití napěťové rampy minimálně jeden krok mezi počátečním a koncovým napětím. V případě, že se budeme snažit nastavit na přístroji hodnoty, které překračují dané meze přístroje, bude použita nejbližší mez a do zásobníku chyb se uloží chyba indikující překročení meze.

Seznam použitelných příkazů pro tento přístroj je v příloze č. I.

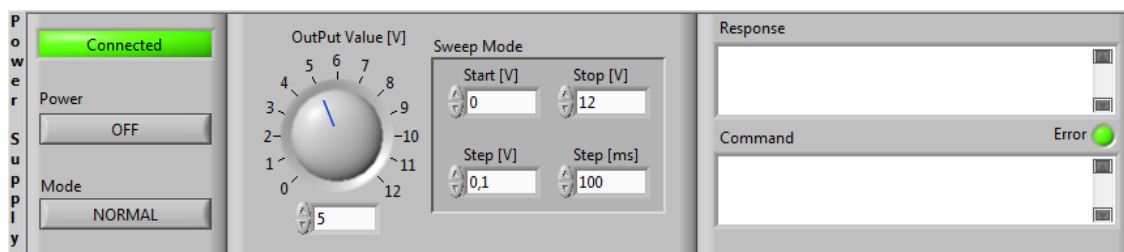
Specifikace požadavků na zdroj napětí:

- Zapnutí/vypnutí přístroje
- Velikost výstupního napětí 0 V až 12.0 V
- Napěťová rampa (Sweep)
 - Velikost počátečního a koncového napětí 0 V až 12.0 V
 - Velikost kroku 0.1 V až 12.0 V
 - Čas mezi kroky 1 mS až 2 s
- Krok změny napětí 0.1V



Obr. 6 Čelní panel zdroje reálného napětí Agilent E3631A

2.1.1 Čelní panel simulátoru zdroje napětí



Obr. 7 Vzhled čelního panelu simulátoru zdroje napětí

Na následujícím obrázku (Obr. 7) je zobrazen čelní panel vytvořeného simulátoru zdroje napětí. Čelní panel je pro přehlednost rozdělen do tří částí.

V levém kraji obsahuje dva indikátory pro zobrazení příkazu, který přístroj přijal jako poslední a v případě, že na tento příkaz existuje odpověď, zobrazí se zde také. Také je zde prvek s názvem Error signalizující, zda přijatý příkaz byl rozpoznán a zda hodnoty v něm nepřekračují stanovené meze.

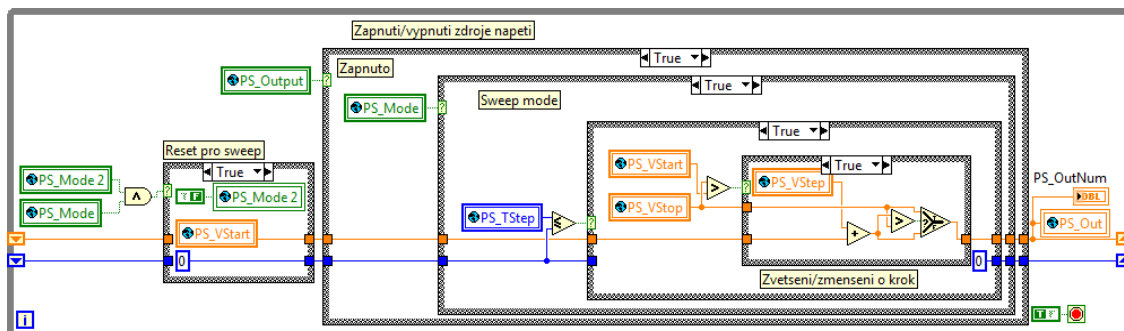
V další části jsou obsaženy prvky pro nastavení amplitudy zdroje napětí a prvky pro nastavení hodnot pro napěťovou rampu.

Poslední část obsahuje dva ovládací prvky, které slouží k přepínání módu a k zapnutí přístroje a jeden indikátor. Tímto indikátorem je znázorněno, zda je k přístroji připojen uživatel.

Aktuálně nastavené hodnoty jsou zobrazeny na ovládacích prvcích. Hodnoty z čelního panelu se ukládají do globálních proměnných, které jsou dále využity jako parametry pro simulaci funkce přístroje.

2.1.2 Kód simulátoru zdroje napětí

Program simulující zdroj napětí je vytvořen pomocí opakující se smyčky. Toto subVi je voláno v hlavní části programu každou 1mS. K ovládání tohoto zdroje napětí jsou využity globální proměnné. V těchto proměnných jsou uloženy veškeré informace o aktuálním nastavení přístroje. První důležitý krok co se v programu provede je kontrola, zda je přístroj zapnutý. V případě, že ne je na výstup přivedena hodnota 0. Pokud je přístroj zapnutý, je na výstupu hodnota odpovídající zvolenému módu zdroje napětí.



Obr. 8 Zdrojový kód programu simulující zdroj napětí

Zdroj napětí má dle požadavků dva módy. V prvním je na výstup přivedena stála hodnota odpovídající hodnotě nastavené pomocí ovládacího panelu. V druhém módu je na vstupu skokově měněná hodnota. Zdrojový kód pro druhý mód je na obrázku 8. V tomto režimu je kontrolováno, zda aktuální hodnota již dosáhla konečné hodnoty. V případě, že ne je a uplynul čas, po který má být aktuální hodnota na výstupu, je k aktuální hodnotě přičtena nebo odečtena hodnota odpovídající velikosti kroku. Tato smyčka je provedena vždy pouze jednou a to v případě, že je zavolána hlavním programem.

2.2 Simulátor generátoru funkcí

Generátor funkcí, jako další přístroj výsledného simulátoru, slouží k simulaci různých typů signálu. Jako výstup z generátoru funkcí může být signál sinusový, obdélníkový, trojúhelníkový nebo signál ve tvaru pily. Tyto výstupní signály je možno doplnit o šum. Také u toho přístroje je možnost sweep módu, to je takový signál, u kterého postupně narůstá frekvence z výchozí hodnoty ke koncové po zadaných krocích. Velikost těchto kroků, stejně jako počáteční a koncovou hodnotu a časový odstup mezi jednotlivými kroky, zadává uživatel.

Uživatel může zvolit frekvenci signálu od 1Hz až po hodnotu 100 Hz, amplitudu výstupního signálu lze měnit v rozsahu 0 V po 10 V. K tomu výstupní signálu je možno přidat šum o velikosti amplitudy od 0.01 V až po velikost 2 V. Také je zde možnost využití frekvenční rampy, kdy se v určitých časových intervalech, po zadaném kroku, mění frekvence výstupního signálu.

Seznam použitelných příkazů pro tento přístroj je v příloze č. II.

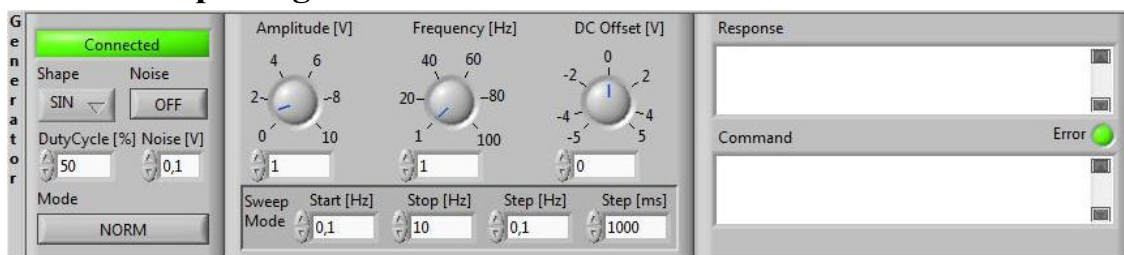
Specifikace požadavků na generátor funkcí:

- Výběr typu průběhu generovaného signálu (sinusový, obdélníkový, trojúhelníkový, pila, plus možnost přidat šum)
- Volba velikosti DC offsetu
- Frekvenční rampa (Sweep)
 - Velikost počáteční a koncového frekvence 1 Hz až 100 Hz
 - Velikost kroku 1 Hz až 100 Hz
 - Čas mezi kroky 1 mS až 1 s
- Amplituda výstupního signálu 0 V až 10 V
- Krok změny amplitudy signálu 0.1 V
- Amplituda šumu 0V až 2V
- Krok změny amplitudy šumu 0.01V
- Frekvence výstupního signálu 1 Hz až 100 Hz
- Krok změny frekvence 1 Hz



Obr. 9 Čelní panel reálného generátoru funkcí Agilent 33220A

2.2.1 Čelní panel generátoru funkcí



Obr. 10 Čelní panel simulátoru generátoru funkcí

Na obrázku (Obr. 10) je zobrazen čelní panel vytvořeného simulátoru generátoru funkcí. Čelní panel je pro přehlednost rozdělen do tří částí.

V pravé krajní části jsou indikátory pro zobrazení posledně přijatého příkazu a odeslané odpovědi na tento příkaz. Tyto dva indikátory jsou doplněny o prvek signalizující (Error), zda je rozpoznán přijatý příkaz a zda hodnoty v něm obsažené nepřekračují definované meze.

V prostřední části jsou prvky pro nastavení většiny hodnot generátoru, jakou jsou amplituda, frekvence a DC offset signálu. Také zde jsou prvky k nastavení hodnot pro frekvenční rampu.

V poslední části jsou prvky pro výběr typu signálu, nastavení střidy (využitelné pouze u signálu s obdélníkovým průběhem) a výběr módu generátoru. Tato část také obsahuje prvek povolující přidání šumu a prvek, jímž můžeme nastavit velikost šumu. Tato poslední část také obsahuje signalizaci, zda je k přístroji uživatel připojen.

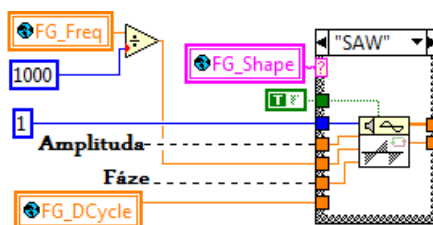
Na tlačítkách čelního panelu jsou viditelné aktuálně nastavené hodnoty. Hodnoty z čelního panelu se ukládají do globálních proměnných, které jsou dále využity jako parametry pro simulaci funkce přístroje.

2.1.2 Kód simulátoru generátoru funkcí

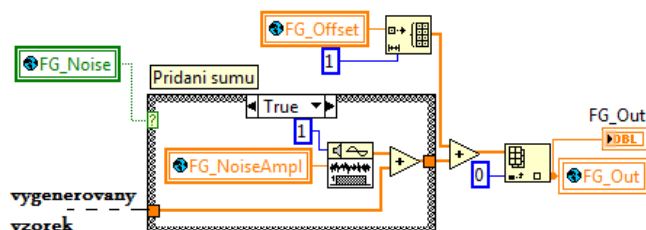
Program generátoru je vytvořen tak, že při každém průběhu smyčky je vygenerován jeden vzorek výsledného signálu. Maximální frekvence signálu je definována na 100 Hz. Vzhledem k tomu, že subVi s funkčním blokem generátoru je voláno 1 ms intervalech je rozestup mezi vzorky 1 ms, což by dle Shanonnova vzorkovacího teorému bylo postačující až pro signály dosahující 500 Hz. Z důvodu hladkého běhu aplikace je maximální frekvence ponechána na 100 Hz. Čímž získáme 10 vzorků na jednu periodu signálu.

Samotný program simulující funkci generátoru, lze rozdělit do několika částí. Na hlavní část, kde se generuje samotný signál o pevné nebo signál o proměnné frekvenci („sweep“ mód). Dále část pro přidání šumu k signálu a část pro stejnosměrný napěťový offset výsledného signálu.

Ke generaci vzorků signálu bylo využito bloků, které nabízí přímo programovací prostředí LabView. Bylo využito bloků pro generaci těchto typů průběhů signálů: sinusový, obdélníkový, trojúhelníkový a pilovitý průběh.

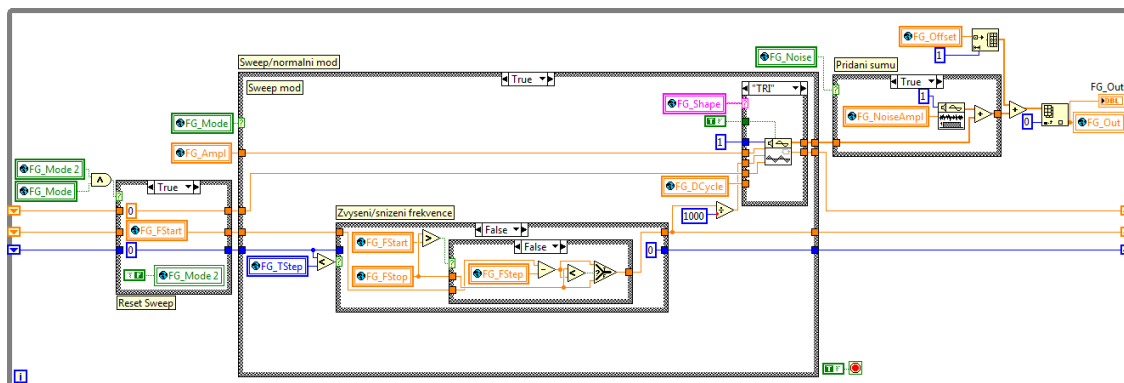


Obr. 11 Část programu pro vygenerování jednoho vzorku signálu



Obr. 12 Část programu pro přičtení šumu a DC offsetu k signálu

Takto zkonstruovaný generátor splňuje všechny požadavky, které byly požadovány při definování vlastností přístrojů.



Obr. 13 Výsledný zdrojový kód generátorů funkcí

2.3 Simulátor digitálního multimetru

Třetím přístrojem, který bude obsažen ve výsledném simulátoru, je digitální multimetr. Tento multimetr má přepínatelný vstup. Vstup lze přepnout mezi výstupem ze zdroje napětí a výstupem z generátoru funkcí. Tento multimetr může měřit signál, jenž je přiveden na vstup multimetru, velikost efektivní hodnoty, velikost amplitudy střídavé složky, velikost stejnosměrné složky, dále pak umožňuje změřit tzv. klouzavé minimum nebo maximum, a také frekvenci signálu. U přístroje je možno zvolit napěťový rozsah. Na výběr jsou tři rozsahy 1V, 2V a 20V. Digitální multimetr měří zadané hodnoty z půl vteřinového souboru vzorků.

Seznam použitelných příkazů pro tento přístroj je v příloze č. III.

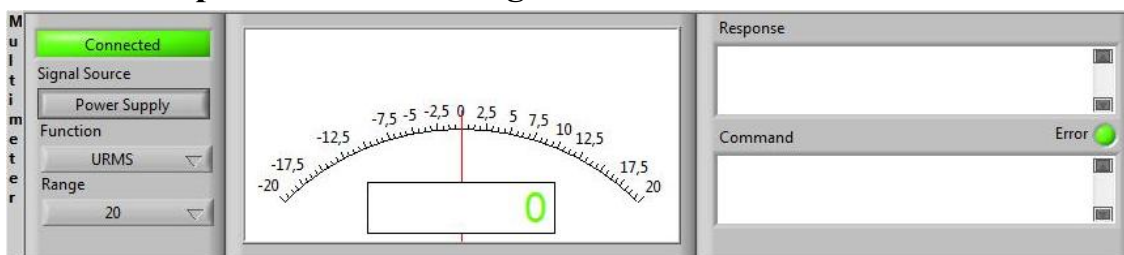
Specifikace požadavků na digitální multimetr:

- Výběr měřené veličiny (U_{RMS} , U_{AC} , U_{DC} , U_{MAX} , U_{MIN} , f)
- Přepnutí zdroje měřeného signálu (funkční generátor / zdroj napětí)
- Výběr rozsahu měřeného napětí



Obr. 14 Čelní panel reálného digitálního multimetru Agilent 34410A

2.3.1 Čelní panel simulátoru digitálního multimetru



Obr. 15 Čelní panel simulátoru digitálního multimetru

Na obrázku výše (Obr. 15) je zobrazen čelní panel digitálního multimetru. Ten je také rozdělen do 3 částí.

V pravé krajní části jsou indikátory pro zobrazení posledně přijatého příkazu a odeslané odpovědi na tento příkaz. Tyto dva indikátory jsou doplněny o prvek signalizující, zda je rozpoznán přijatý příkaz a zda hodnoty v něm obsažené nepřekračují definované meze.

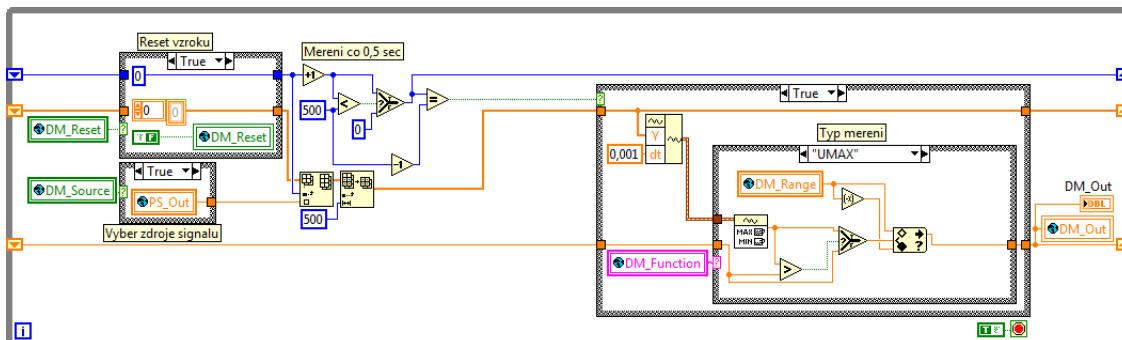
V prostřední části je ručičkový indikátor, zobrazující aktuálně změřenou hodnotu. Tento ukazatel je doplněn o digitální indikátor, pro jednodušší přečtení změřené hodnoty.

V levé části jsou ovládací prvky pro digitální multimetr. Nachází se zde tlačítko pro výběr zdroje signálu k měření, nabídka pro výběr měřené hodnoty a nabídka pro volbu rozsahu multimetru. Také je zde indikátor pro signalizaci toho, zda je k multimetru připojen uživatel.

Na tlačítkách čelního panelu jsou viditelné aktuálně nastavené hodnoty. Hodnoty z čelního panelu se ukládají do globálních proměnných, které jsou dále využity jako parametry pro simulaci funkce přístroje.

2.1.2 Kód simulátoru digitálního multimetru

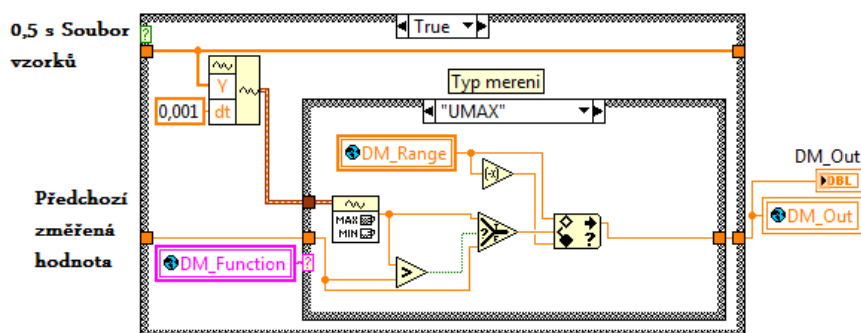
Funkční blok digitálního multimetru je vytvořen ve smyčce, která je cyklicky volána hlavním programem simulátoru. Po zavolání této smyčky se její obsah provede pouze jednou.



Obr. 16 Výsledný zdrojový kód digitálního multimetru

Na začátku tohoto subVI vybrán zdroj signálu ze kterého se odebírají vzorky, protože subVI s funkcí digitálního multimetru je voláno co 1 ms je rozestup mezi sbíranými vzorky 1 ms. Část programu zobrazující vytvoření souboru vzorků je zobrazena na **Chyba! Nenalezen zdroj odkazů..**

Po vytvoření půl vteřinového souboru vzorků je tento soubor změřen pomocí vybrané funkce a na výstup digitálního multimetru je zapsána změřená hodnota. Část subVI, kde je zobrazen blok pro měření vybrané hodnoty a její uložení na výstup digitálního multimetru a globální proměnné je na **Chyba! Nenalezen zdroj odkazů..** Tato hodnota je na výstupu uložena do chvíle než se změří nový soubor vzorků, kdy je stará změřená hodnota přepsaná novou. K měření jednotlivých hodnot je využito funkcí, které nabízí prostředí LabVIEW.



Obr. 17 Část programu, obsluhující měření zvolené hodnoty

2.4 Simulátor osciloskopu

Posledním přístrojem, který bude obsažen ve výsledném simulátoru, je osciloskop. Tento přístroj má také přepínatelný vstup. Vstup lze přepnout mezi výstupem ze zdroje napětí a výstupem z generátoru funkcí. Osciloskop má nastavitelný napěťový rozsah vstupu, který lze nastavit v rozmezí od 0.1 V/dílek do 10 V/dílek. Konkrétní hodnoty pro nastavení napěťového rozsahu jsou uvedeny v tabulce 9.

V/div
0.1
0.5
1
2
5
10

Tab. 9 Napěťové rozsahy pro osciloskop

Dále je možno nastavit velikost časové základny v rozmezí od 100 ms do 1 s. Dále zde je možnost přepínání módu pro single měření a kontinuální měření. Další vlastností osciloskopu je přepínání módu AUTO/NORM. Poslední vlastností toho přístroje je nastavení DC offsetu a to v rozmezí -5 V až +5 V po kroku 0,1 V.

Seznam použitelných příkazů pro tento přístroj je v příloze č. IV.

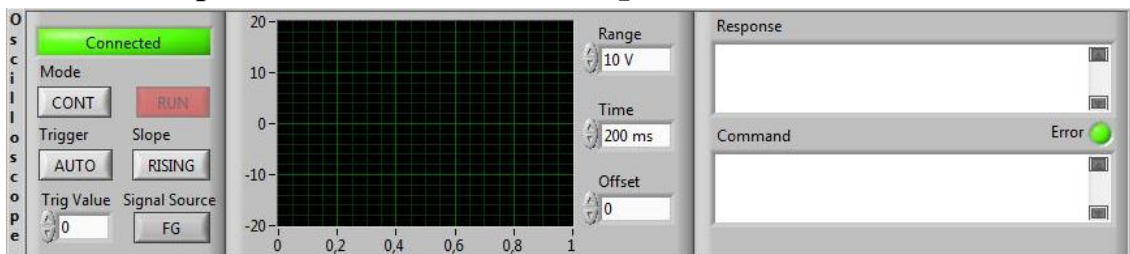
Specifikace požadavků na osciloskop:

- Volba velikosti časové základny
- Volba rozsahu vstupu
- Volba velikosti DC offsetu
- Přepínání typu měření single/kontinuální
- Přepínání trigger módu AUTO/NORM



Obr. 18 Čelní panel reálného osciloskopu Agilent MSO6012A

2.4.1 Čelní panel simulátoru osciloskopu



Obr. 19 Čelní panel simulátoru osciloskopu

Na obrázku 19 je zobrazen čelní panel simulátoru osciloskopu. Čelní panel tohoto přístroje je rozdělen jako ostatní do tří částí.

V pravé krajní části jsou indikátory pro zobrazení posledně přijatého příkazu a odeslané odpovědi a je zde také indikátor pro informaci o tom, zda přijatý příkaz byl rozpoznán a zda obsahoval hodnoty, jež jsou v daných mezích.

V prostřední části je obrazovka osciloskopu s prvky přímo ovlivňujícími zobrazení naměřených hodnot.

V levé části jsou prvky přímo ovlivňující měření. Je zde možnost přepnutí režimu osciloskopu z kontinuálního měření na měření jednorázové. Dále jsou zde prvky nastavující mód triggru a jeho vlastnosti. A také je zde indikátor pro signalizaci toho, zda je k osciloskopu připojen uživatel.

Na tlačítkách čelního panelu jsou viditelné aktuálně nastavené hodnoty. Hodnoty z čelního panelu se ukládají do globálních proměnných, které jsou dále využity jako parametry pro simulaci funkce přístroje.

2.1.2 Kód simulátoru osciloskopu

Vytvoření toho subVi bylo z hlediska časové náročnosti a pracnosti největší. Avšak výsledný simulátor osciloskopu splňuje všechny požadavky, které byly vyřčeny při definici vlastností tohoto přístroje.

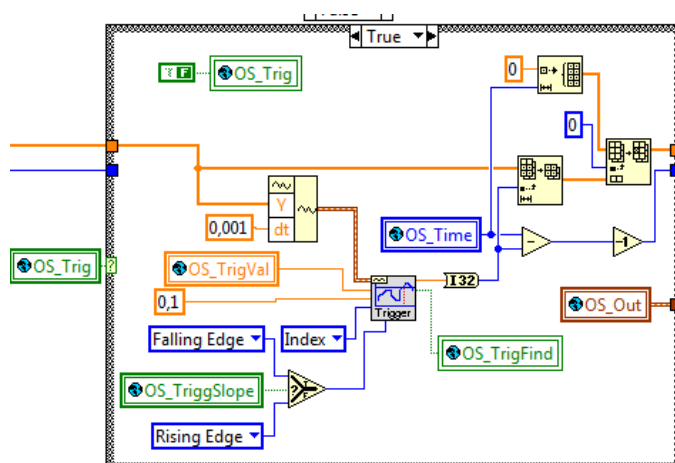
Funkční blok osciloskopu je vytvořen smyčkou, která provede po každém jejím zavolání jeden průběh kódu umístěn uvnitř této smyčky. Program je rozdělen strukturou CASE na dvě části. První část obstarává kontinuální měření a v druhé části je kód pro jednorázové změření průběhu signálu na vstupu.

V závislosti na nastavené časové konstantě je vyčítán určitý počet vzorků, z kterého je následně poskládán obraz na obrazovce osciloskopu (Tab. 10). Do chvíle než je potřebný počet vzorků nasbírán je na obrazovce zobrazen poslední průběh, ale pouze v případě, že je trigger nastaven v režimu AUTO. V opačném případě není na obrazovce zobrazen žádný průběh.

Časová konstanta	Počet vzorků
20 ms/div	100
100ms/div	500
200ms/div	1000

Tab. 10 Závislost počtu vzorků na zvolené časové konstantě

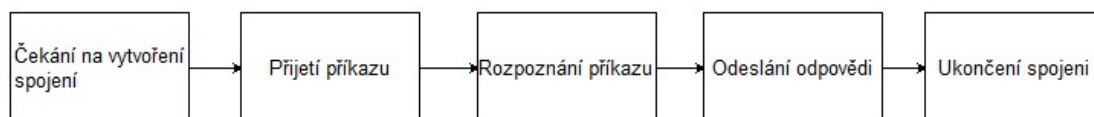
Pro vytvoření funkce triggeru byl využit blok přístupný v programovacím prostředí LabView. Ten v případě nalezení hodnoty odpovídající triggerovací úrovni vrátí umístění v poli kdy je triggerovací podmínka splněna. Následně se do globální proměnné uloží příznak, že triggerovací podmínka byla splněna a pole vzorků se posune na pozici udanou tímto blokem a provede se dovzorkování signálu tak, aby odpovídal počet vzorků počtu potřebných pro zobrazení průběhu signálu na obrazovce osciloskopu. V případě, že trigger nastaven v módu NORM a není nalezen bod kdy je triggerovací podmínka splněna není na obrazovce osciloskopu zobrazen žádný průběh a obrazovka zůstává prázdná do chvíle než je tato podmínka splněna nebo je změněn mód na AUTO. Část kódu obsluhující nalezení triggerovací podmínky a úpravy pole vzorků je zobrazena na obrázku níže (Obr. 20).



Obr. 20 Část kódu osciloskopu pro nalezení bodu splnění triggerovací podmínky

3 Komunikace simulovaný přístroj – uživatel

Ke komunikaci mezi simulátorem a programem, který slouží k obsluze simulátoru, bylo vybráno rozhraní Ethernet. Z důvodů nejjednoduššího provedení komunikační smyčky, která nepotřebuje žádné hardwarové části, jako by to mu bylo například u RS-232 nebo GPIB. Je zde využita lokální IP adresa, díky které můžeme komunikovat s každým z navržených přístrojů samostatně, díky tomu, že každý simulovaný přístroj využívá ke komunikaci rozdílný port. V případě RS-232 je nemožné vyřešit komunikaci tak, abychom mohli s každým přístrojem komunikovat zvlášť, a to z důvodu velkých požadavků na hardwarové vybavení počítače. K tomu by byla potřeba šestnácti jednotlivých RS-232 rozhraní (pro každý přístroj 2, jeden na straně PC a druhý, jakoby na straně simulovaného zařízení), čímž běžné počítače nedisponují.

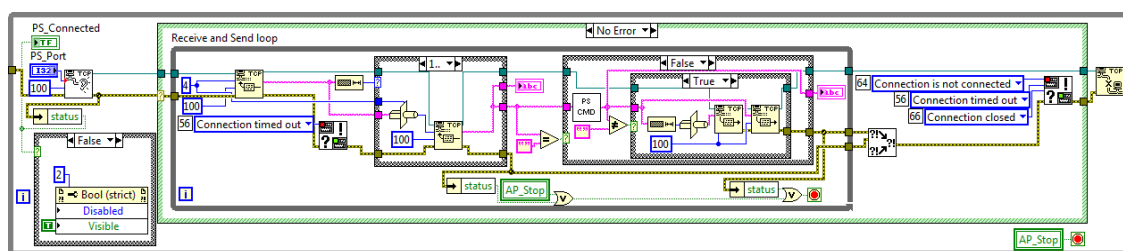


Obr. 21 Blokové schéma komunikační smyčky na straně simulátoru

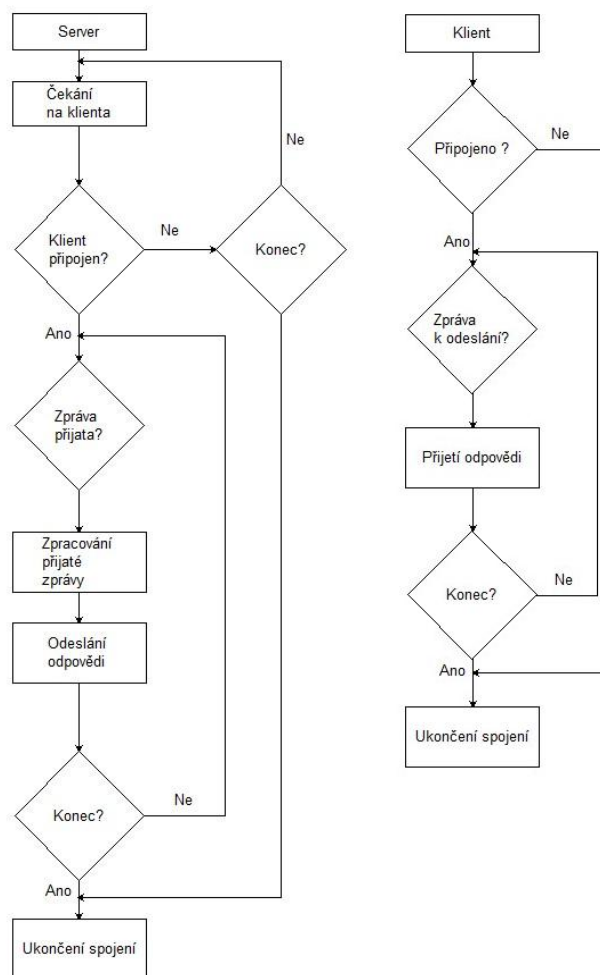


Obr. 22 Blokové schéma komunikační smyčky na řídicí aplikaci

Komunikace mezi přístroji by se dala rozdělit na dvě části. První částí je komunikace na straně samotného simulátoru a tou druhou částí je komunikace na straně programu obsluhující tento simulátor.



Obr. 23 Část programu obsluhující komunikaci na straně simulátoru



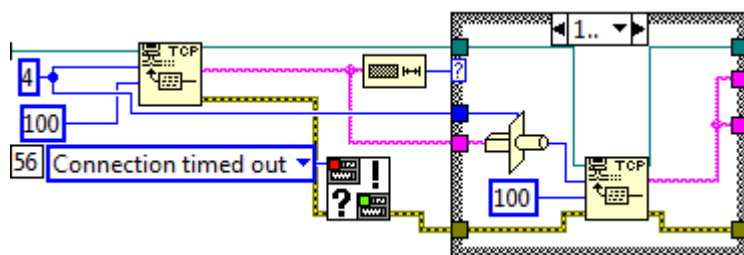
Obr. 24 Vývojový diagram komunikace

Simulátor při komunikaci funguje jako server. V počátku server vyčkává na vytvoření komunikačního kanálu. Doba, kterou server vyčkává na vytvoření komunikačního kanálu, je nastavena na 100 ms. Pokud nedojde k vytvoření komunikačního kanálu do této doby, proběhne celá smyčka a začne se znovu vyčkávat na vytvoření komunikačního kanálu.

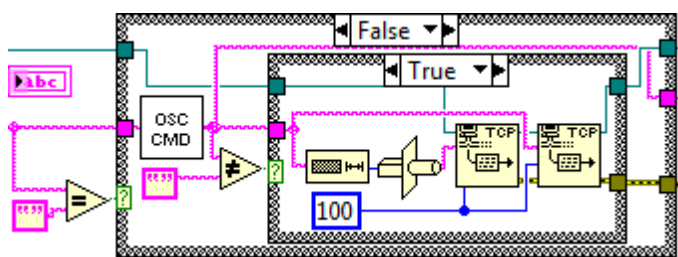
Po té co se vytvoří komunikační kanál, dojde nejprve k přijetí zprávy. První 4 byty zprávy určují délku zprávy, a také to, zda byla přijata zpráva neobsahující prázdnou množinu znaků. Pokud je velikost zprávy různá od nuly, je uvedená zpráva o této délce přijatá a následně poslaná do další části programu, která rozpoznává přijatou zprávu a v případě, že přijatá zpráva byla rozpoznána jako dotaz, vytvoří požadovanou odpověď na tento konkrétní dotaz. Jestliže přijatou zprávou byl příkaz, který nevrací žádnou odpověď, je do proměnné pro odpověď vložen prázdný řetězec.

Následně probíhá komunikace opačným směrem. Nejdříve je zkontrolováno, zda se program nepokouší poslat prázdnou zprávu. V případě, že ano, přeskočí se v programu posílání odpovědi a celý běh programu se opakuje znovu od bodu přijetí zprávy. V případě, že se nejedná o prázdnou zprávu, tak je tato zpráva nejdříve doplněna o 4 byty určující její délku, a pak je odeslána na zadanou adresu a port.

Na následujících obrázcích (Obr. 25 a Obr. 26) jsou zobrazeny části programu ze simulátoru sloužící pro přijetí příkazu a odeslání odpovědi.



Obr. 25 Část kódu obsluhující přijetí zprávy



Obr. 26 Část kódu obsluhující posílání zprávy

Aby nedošlo k uvíznutí programu při čekání na přijetí zprávy, je zde nastaven čas, po který se čeká na přijetí zprávy na 100ms. Poté se blok pro přijetí zprávy přeskočí a tím se zabráni velkému zdržení programu. Tento čas je také nastaven u odesílání odpovědi. To z důvodů, aby se při nenadálém problému s odesíláním odpovědi program nezasekl na dlouhou dobu v bloku pro odesílání zprávy. Za oběma částmi je vloženo subVi, které upraví výstupní chybu na varování. Tento blok je použit, aby nedošlo k ukončení běhu celého programu z důvodu chyb při komunikaci.

Program pro vzdálenou obsluhu simulátoru funguje jako klient, jenž se k simulátoru (serveru) připojuje a vytváří tak komunikační kanál. Průběh komunikace z hlediska klienta je téměř shodný se serverem. Jediný rozdíl je ten, že klient nevyčká na vytvoření spojení, ale sám jej vytváří.

Každý přístroj má svou vlastní komunikační smyčku, která běží nezávisle na ostatních částech programu. Ke komunikaci byly využity porty, které jsou znázorněny v následující tabulce (Tab. 11). Porty jsou zvoleny, tak aby nedocházelo k případné kolizi s běžnými aplikacemi.

Přístroj	Číslo portu
Zdroj napětí	9999
Generátor funkcí	9998
Digitální multimetr	9997
Osciloskop	9996

Tab. 11 Porty určené pro komunikaci s přístroji

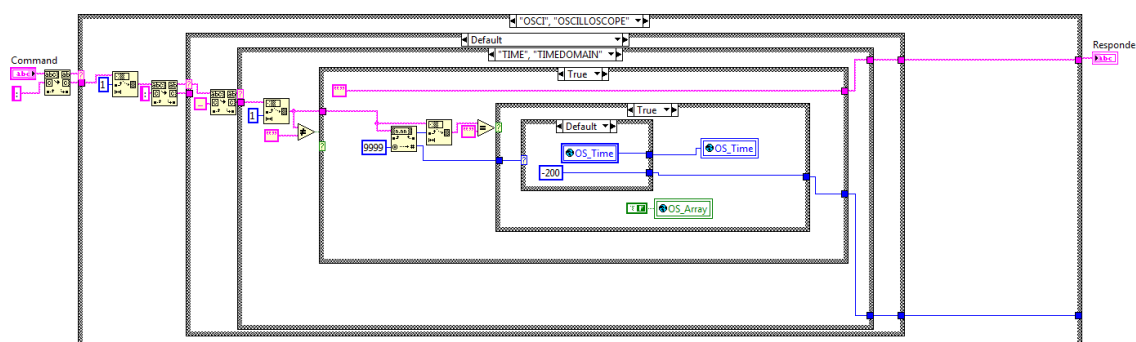
Pro názornost je na dalším obrázku (**Chyba! Nenalezen zdroj odkazů.**) zobrazen celý blok programu obsluhující komunikaci ze strany simulátoru přístroje. Tento blok je shodný pro všechny čtyři přístroje.

4 Rozpoznání příkazů pro dálkové ovládání přístrojů

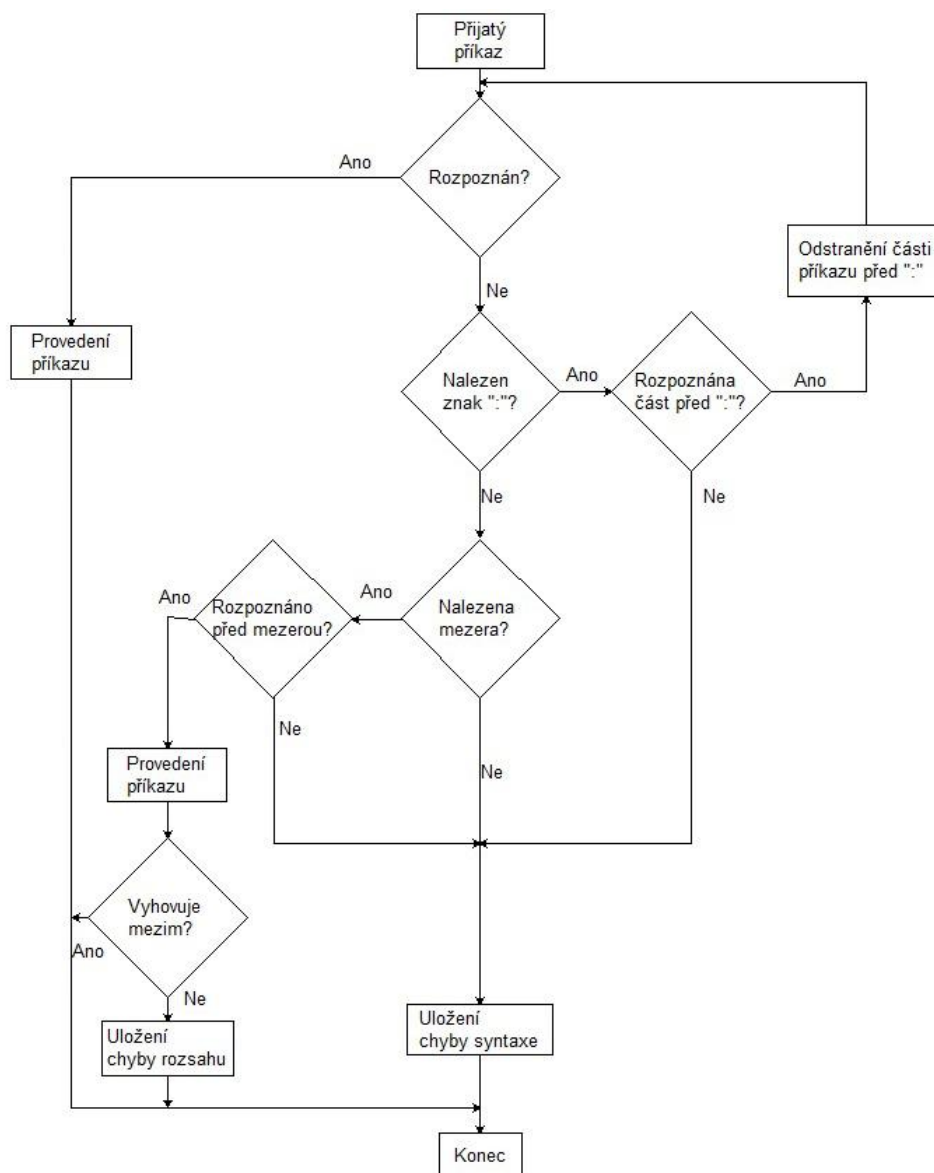
Pro případ, kdy by byl požadavek komunikovat s vytvořeným simulátorem pomocí RS 232 je do příkazu, vložena na začátek část určující přístroj, pro který je následující část příkazu určena a proto, že při komunikaci pomocí sériového rozhraní nelze rozlišovat komunikační port, jako tomu je v případě komunikace pomocí rozhraní Ethernet. Kde by toto rozlišování bylo nadbytečné.

Další důležitou částí, potřebnou pro komunikaci se simulátorem pomocí příkazů je subVi, které slouží k rozpoznání přijatého příkazu. Všechny příkazy mají jasně definovanou strukturu. V první části příkazu je zakódován přístroj, pro který je příkaz určený, tato část je oddělena dvojtečkou a dále následují další části specifikující funkci příkazu. Tyto části jsou od sebe také oddělovány dvojtečkou. V případě, že příkaz nastavuje v přístroji nějakou hodnotu, je tato hodnota oddělena od příkazu mezerou.

Jako příklad může posloužit příkaz pro nastavení časové základny osciloskopu. Celý příkaz je ve tvaru „OSCI:TIME 200“. V bloku pro rozpoznání příkazu se nejprve vyhledá dvojtečka a celý příkaz se rozdělí na dvě části to je na „OSCI“ a „:TIME 200“. Poté se část příkazu, která je před dvojtečkou, přivede na vstup CASE struktury. Potom co je první část příkazu rozpoznána a nalezena, se u druhé části odstraní dvojtečka a poté se zkusí vyhledat znovu dvojtečka. Toto se několikrát opakuje, dokud je v zbytku příkazu nalezen symbol „:“. Pokud měl příkaz správnou syntaxi a je to příkaz obsahující hodnotu k nastavení vyhledá se v něm mezera oddělující samotné tělo příkazu od hodnoty. Tím je vzorový příkaz rozdělen na „TIME“ a „ 200“. U části obsahující nastavovanou hodnotu je odebere mezera a je, pokud se jedná o číslo převedena z textového řetězce na číslo požadovaného formátu. Následně je tato hodnota zkontrolována, zda vyhovuje definovaným mezím a uloží se do globální proměnné. V případě, že se za příkazem vyskytnou další znaky, je tato situace vyhodnocena jako špatný příkaz



Obr. 27 Kód subVi pro rozpoznání příkazu



Obr. 28 Vývojový diagram rozpoznání příkazu

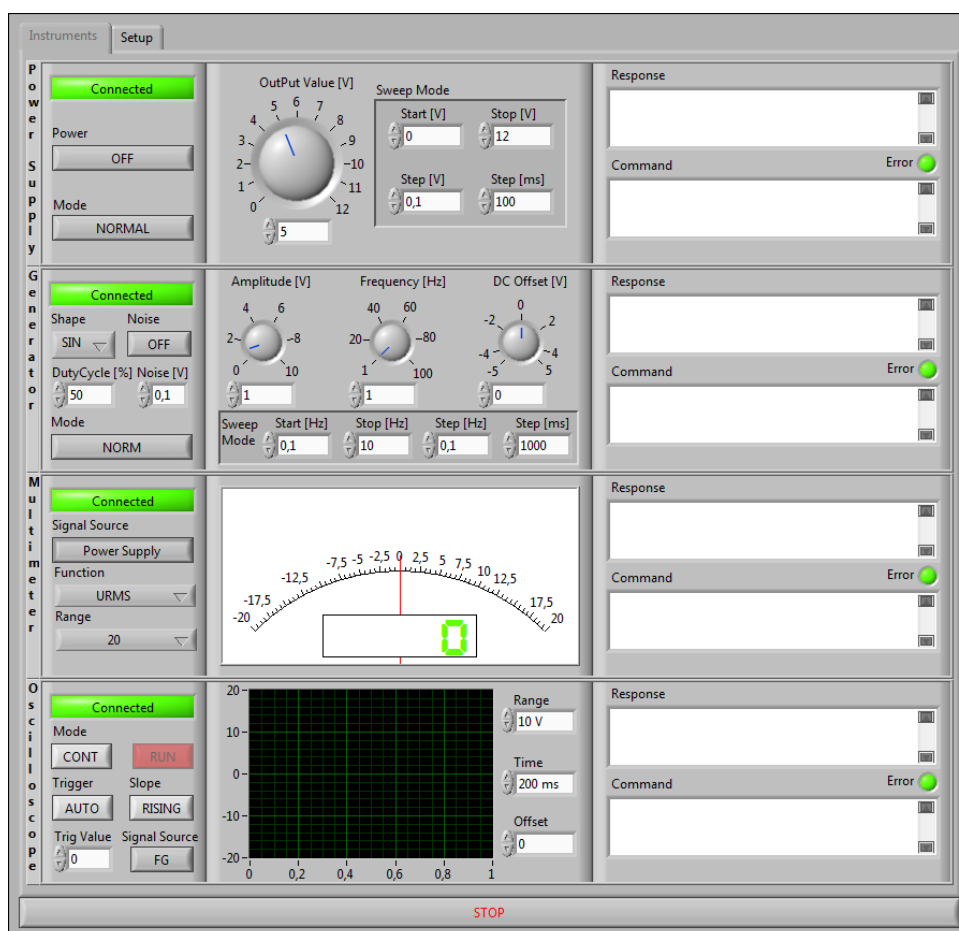
Pokud se jedná o příkaz vracející odpověď, která je složena z hodnoty uložené v globální proměnné. Je nejdříve hodnota z globální proměnné vyčtena a předvedena do textové podoby a poté je tato hodnota poslána na výstup subVi pro zpracování příkazu.

V případě, že vše proběhne v pořádku, není do globální proměnné, sloužící jako zásobník chyb, vložena žádná hodnota. V případě, není příkaz rozpoznán, je do zásobníku chyb uložena hodnota -100, která signalizuje, že příkaz není rozpoznán. V situaci kdy je příkaz rozpoznán, ale hodnota, kterou uživatel tímto příkazem chce nastavit je mimo meze přístroje je do zásobníku chyb uložena hodnota -200 a do globální proměnné se uloží nejbližší hodnota vyhovující mezím přístroje.

Zásobník chyb nemá limitován počet hodnot, které lze do něj uložit. A chyby se v něm kumulují. Ale lze je vymazat příkazem *RST, nebo postupným vyčtením. Při použití příkazu pro vyčtení chyby se vyčte pouze poslední uložená chyba a po vyčtení se ze zásobníku vymaže.

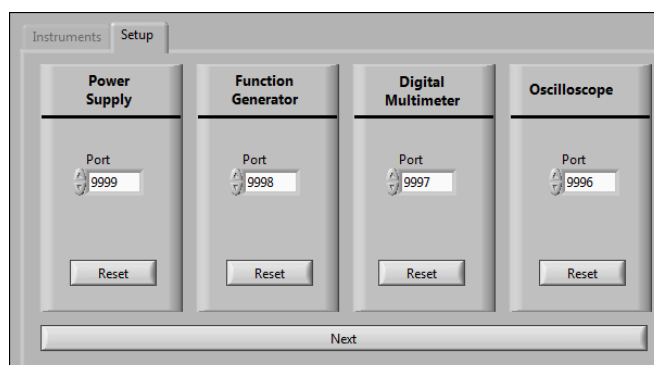
5 Aplikace simulátoru čtyř přístrojů

Grafické rozhraní výsledného programu je složeno ze dvou částí. První část obsahuje čelní panely jednotlivých přístrojů (Obr. 29). Tyto panely jsou detailně popsány u jednotlivých přístrojů v kapitole číslo 2.



Obr. 29 Hlavní obrazovka simulátoru

Druhá část (Obr. 30) slouží k nastavení komunikačního portu pro jednotlivé přístroje a obsahuje také tlačítko pro reset přístrojů. Tímto tlačítkem uvedeme přístroj do defaultního stavu a vynulujeme zásobník chyb.



Obr. 30 Část obrazovky obsahující nastavení přístrojů

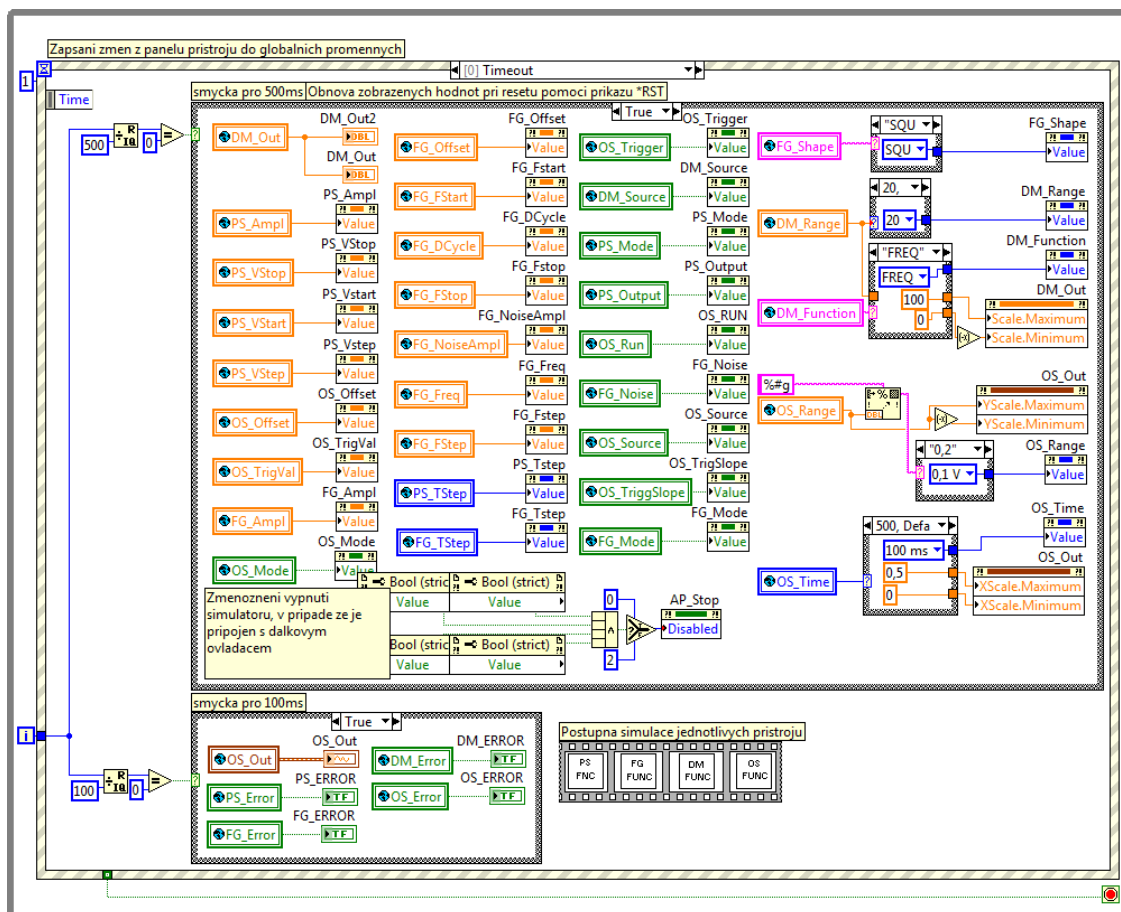
Pro ukončení běhu simulátoru je ve spodní části tlačítko stop, které ukončí všechny přístroje a vypne simulátor.

Při spuštění simulátoru je jako první zobrazena obrazovka s nastavením, kde může uživatel nastavit komunikační porty jednotlivých přístrojů. Na hlavní obrazovku s čelními panely jednotlivých přístrojů se uživatel dostane pomocí tlačítka s popiskem „Next“. Do obrazovky s nastavením portů a tlačítkem pro resetování přístrojů, je uživateli umožněn přístup kdykoliv. Jen možnost změnit port pro komunikaci je v případě, že komunikace již probíhá znemožněna. Ta je opět umožněna ve chvíli, kdy je komunikace s daným přístrojem ukončena.

Reset jednotlivých přístrojů je proveden tak, že se do globálních proměnných uloží defaultní hodnoty a následně se tyto hodnoty zobrazí i na čelním panelu.

Pro jednoduchou synchronizaci a lepší přehlednost byly všechny čtyři přístroje vloženy do jednoho VI. Avšak všechny přístroje se chovají zcela nezávisle na sobě. Při spuštění se jako první činnost provede synchronizace ovládacích prvků s hodnotami uloženými v globálních proměnných. Poté se paralelně rozběhne 5 smyček běžících v 1ms intervalech. Čtyři z pěti těchto smyček obsluhují vytvoření a běh komunikace pro jednotlivé přístroje.

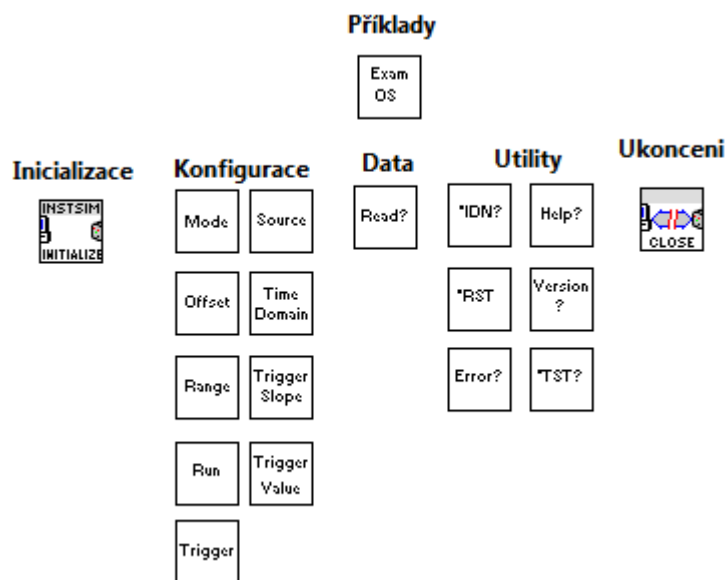
Obsluha čelních panelů a volání jednotlivých subVi obsahující funkce simulovaných přístrojů jsou obsaženy v poslední smyčce. Ve které je také událostmi řízena struktura. Tato struktura reaguje na změnu jakéhokoliv ovládacího prvku na čelních panelech. Tato struktura má nastavenou událost „timeout“, která je vyvolána každou 1 ms, v této události jsou zavolány v definovaném pořadí jednotlivé bloky obsahující simulované přístroje (Obr. 31). V této události jsou vytvořené bloky, které se provedou v intervalech 0,5 s a 0,1 s.



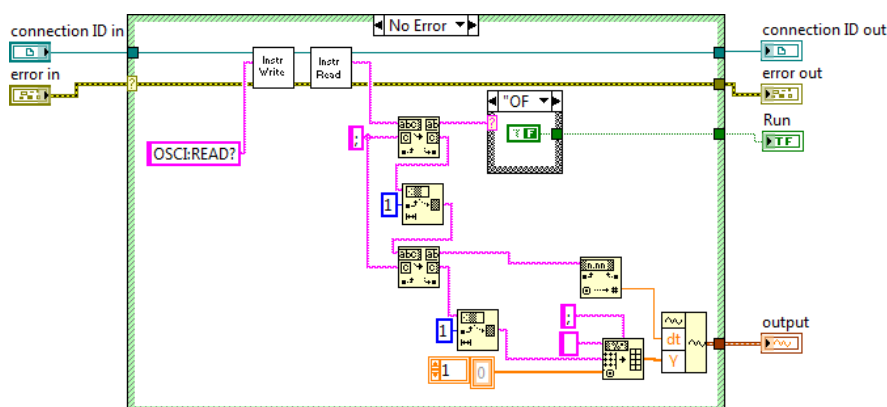
Tato události řízená struktura obsahuje 38 různých událostí, na které tato struktura reaguje. Pro každý z prvků na čelním panelu existuje jedna událost, která daný prvek obsluhuje. Při obsluze jednotlivých prvků je informace v něm obsažena uložena do globální proměnné. V případě, že ovládací prvek ovládá přímo i prvky pro zobrazování dat na čelním panelu, jsou dle obsahu toho ovládacího prvku nastaveny i vlastnosti indikátoru na čelním panelu. Pro příklad je uvedena situace, kdy je změněna hodnota na ovládacím prvku pro výběr velikosti časové konstanty Obr. 32.

6 Přístrojové ovladače pro simulované přístroje

Neméně důležitou částí této diplomové práce bylo i vytvoření přístrojových ovladačů pro jednoduché ovládání jednotlivých přístrojů. Tyto přístrojové ovladače byly vytvořeny pro každý z přístrojů zvlášť. Pro všechny přístroje jsou přístrojové ovladače typu Inicializace, Ukončení a Utility shodné. Rozdílné jsou pouze ovladače typu Konfigurace a Data.



Obr. 33 Příklad souboru jednotlivých přístrojových ovladačů pro osciloskop



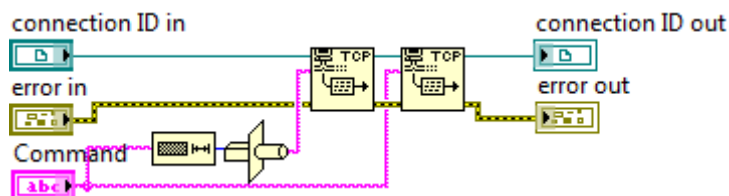
Obr. 34 Příklad přístrojového ovladače pro vyčtení dat změřených osciloskopem

Pro vytvoření přístrojových ovladačů je zapotřebí znát syntaxi jednotlivých příkazů. Avšak pro používání již vytvořených přístrojových ovladačů nutnost znát syntaxi příkazů odpadá. Čímž je vytváření aplikací pro ovládání měřicích přístrojů značně zjednodušeno.

Pro všechny přístroje byly vytvořeny pouze ovladače konfiguruující jednotlivá zařízení, a ovladače sloužící k vyčtení změřených dat z osciloskopu a multimetru. Dalšími ovladači, které byly vytvořeny, jsou ovladače sloužící k identifikaci zařízení, zjištění čísla verze přístroje, resetování přístroje, vyčtení poslední chyby v zásobníku chyb, spuštění otestování přístroje a vypsání helpu.

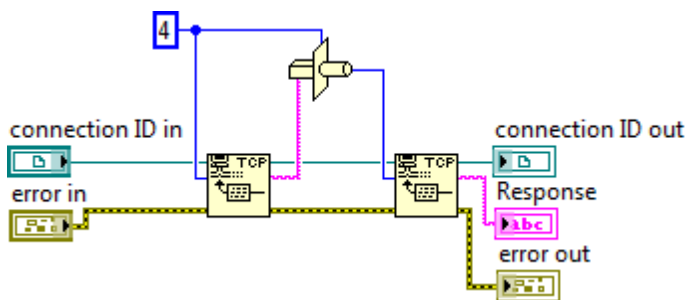
Protože se ve vytvořených ovladačích objevovaly dvě stejné části programu, bylo z těchto dvou částí vytvořené subVi. Tyto subVi výsledně slouží k usnadnění vytváření přístrojových ovladačů. První subVi slouží k úpravě příkazu do formátu v jakém jej lze odeslat. Druhé subVi slouží k přijetí odpovědi ze simulátoru.

Pro odeslání zprávy bylo vytvořeno subVI jehož kód je zobrazen na obrázku 35. Jako první je odeslána délka řetězce obsahující příkaz a až poté je odeslán samotný příkaz.



Obr. 35 Odeslání příkazu

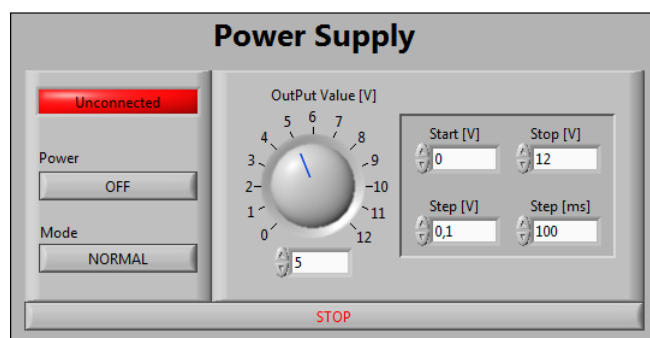
Pro přijetí zprávy bylo vytvořeno subVI jehož kód je zobrazen na obrázku 36. Jako první jsou přijaty 4 Byte, které obsahují informaci o délce zprávy připravené k přijetí.



Obr. 36 Přijetí zprávy

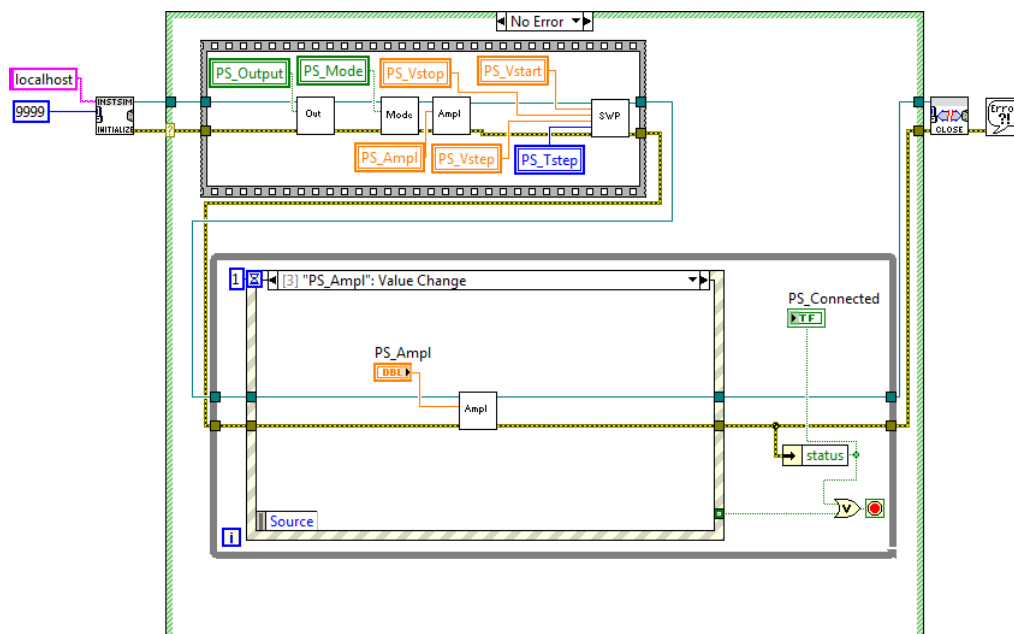
6.1 Příklad vzdáleného ovládání pro zdroj napětí

Vzhled panelu pro vzdálené ovládání vychází z předního panelu samotného přístroje. Jsou zde všechny prvky pro ovládání zdroje napětí. Obsluha těchto ovládacích prvků je provedena pomocí událostmi řízené struktury. Vzhled tohoto panelu je na obrázku 37.



Obr. 37 Vzhled čelního panelu programu pro vzdálené ovládání zdroje napětí

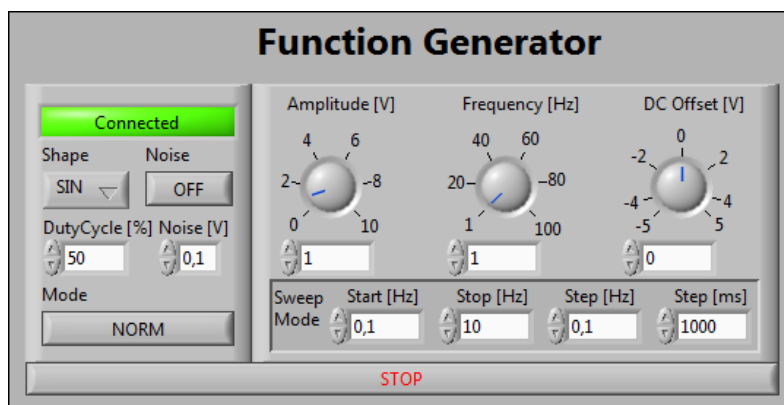
Aby souhlasily hodnoty nastavené na čelním panelu řídicí aplikace, je po vytvoření připojení řídicí aplikace na simulátor provedena synchronizace simulátoru s řídicí aplikací a to, tak že se jednorázově provedou všechny konfigurační přístrojové ovladače. Po počáteční synchronizaci pokračuje program smyčkou, ve které je událostmi řízená struktura. Která obsluhuje veškeré možné události vzniklé změnou hodnot ovládacích prvků na čelním panelu.



Obr. 38 Zdrojový kód programu pro vzdálené ovládání zdroje napětí

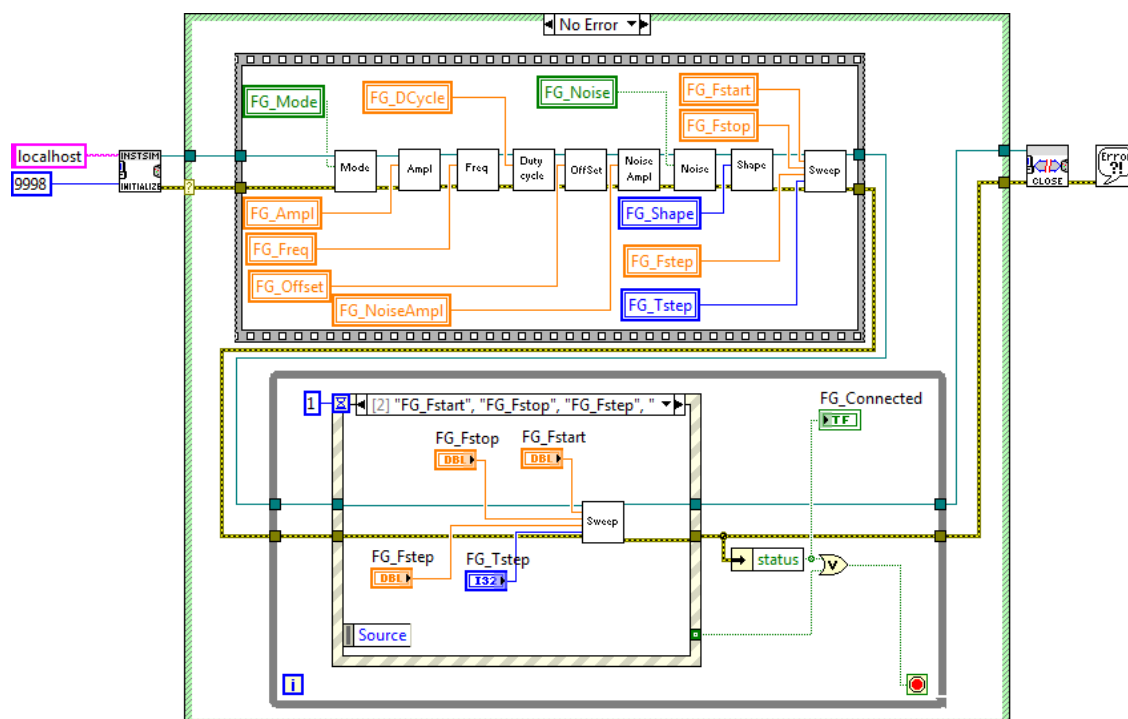
6.2 Příklad vzdáleného ovládání pro generátor funkcí

Vzhled panelu pro vzdálené ovládání vychází z předního panelu samotného přístroje. Jsou zde všechny prvky pro ovládání generátoru. Obsluha těchto ovládacích prvků je provedena pomocí událostmi řízené struktury. Vzhled tohoto panelu je na obrázku 39.



Obr. 39 Vzhled čelního panelu programu pro vzdálené ovládání generátoru

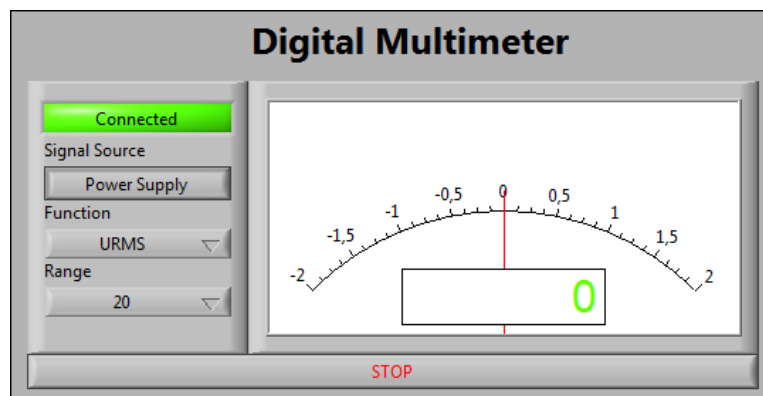
Po vytvoření spojení řídicí aplikace se simulátorem, dojde jako první k synchronizaci hodnot z čelního panelu řídicí aplikace s nastavením simulátoru. Tento krok je proveden pouze jednou po vytvoření spojení. Po této synchronizaci následuje hlavní část programu, která je tvořena smyčkou ve které se nachází událostmi řízená struktura. Tato struktura obsluhuje všechny ovládací prvky na čelním panelu. Při změně hodnoty jakéhokoliv prvky je vyvolána událost, při které se zavolá odpovídající přístrojový ovladač, který novou hodnotu nastaví na simulátoru.



Obr. 40 Zdrojový kód programu pro vzdálené ovládání generátoru

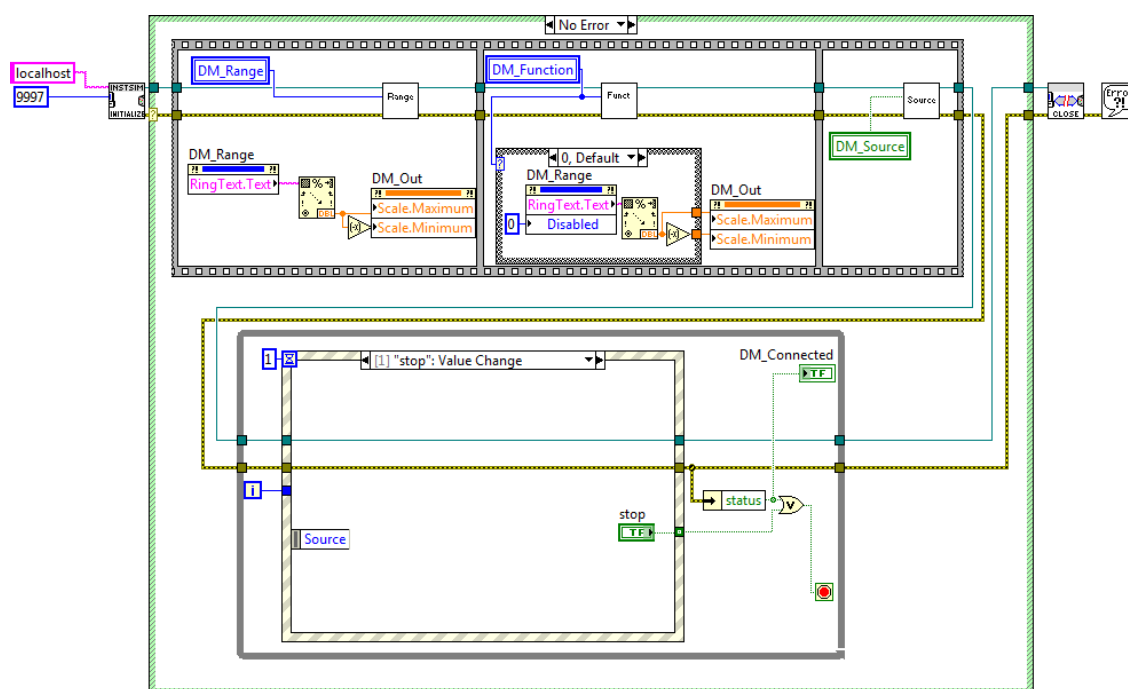
6.3 Příklad vzdáleného ovládání pro digitální multimetr

Vzhled panelu pro vzdálené ovládání vychází z předního panelu samotného přístroje. Jsou zde všechny prvky pro ovládání digitálního multimetru a všechny prvky pro zobrazení změřené hodnoty. Obsluha těchto ovládacích prvků je provedena pomocí událostmi řízené struktury. Vzhled tohoto panelu je na obrázku 41.



Obr. 41 Čelní panel programu pro vzdálené ovládání digitálního multimetru

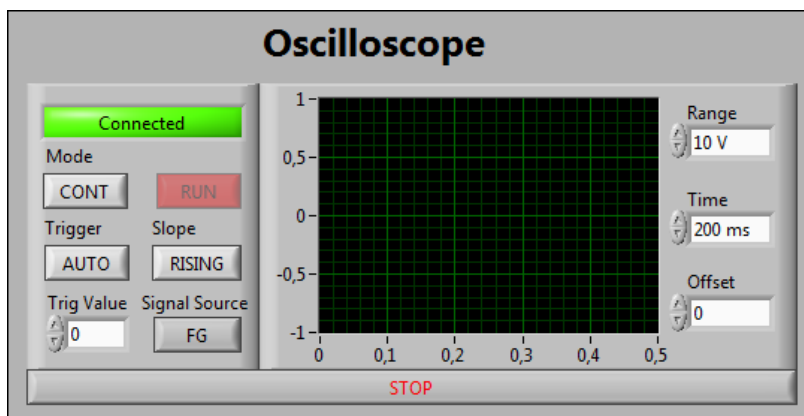
Stejně jako v předchozích případech, i tato řídicí aplikace pro ovládání digitálního multimetru, provádí jako první krok po vytvoření spojení se simulátorem, synchronizaci hodnot na čelním panelu řídicí aplikace s hodnotami nastavenými v simulátoru. Následně program pokračuje smyčkou, ve které je událostmi řízená struktura pro obsluhu všech ovládacích prvků z čelního panelu. Tato struktura má také událost „timeout“, která slouží k vyčtení změřené hodnoty.



Obr. 42 Zdrojový kód programu pro vzdálené ovládání digitálního multimetru

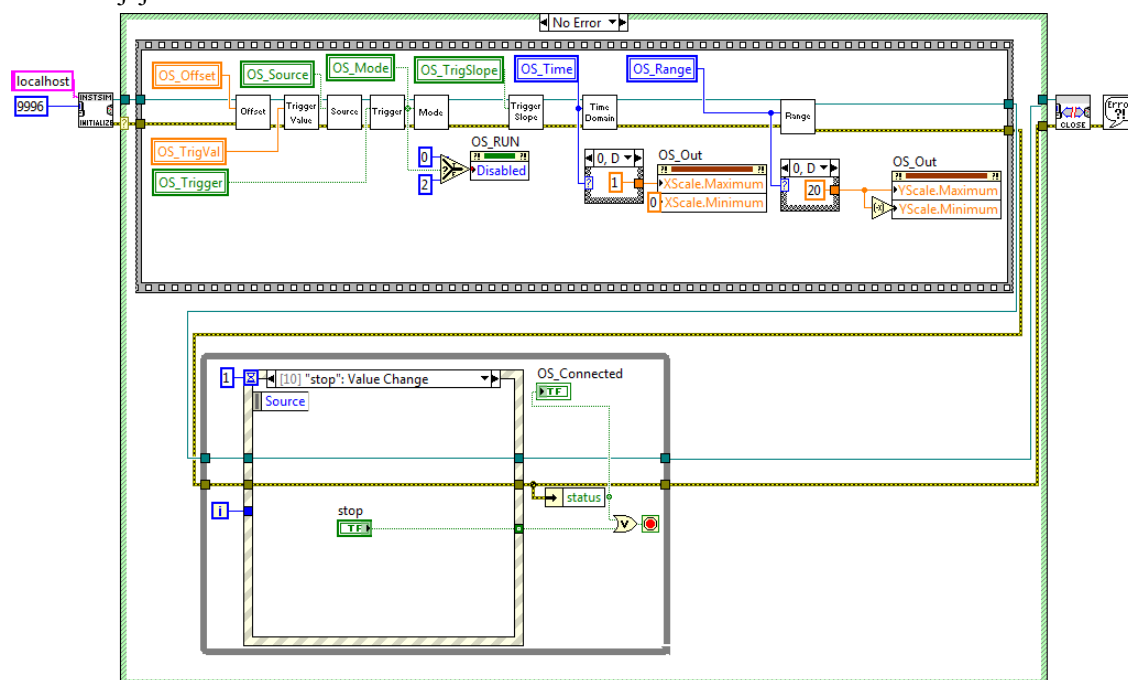
6.4 Příklad vzdáleného ovládání pro osciloskop

Vzhled panelu pro vzdálené ovládání vychází z předního panelu samotného přístroje. Jsou zde všechny prvky pro ovládání osciloskopu a obrazovka sloužící k zobrazení změřeného průběhu. Obsluha těchto ovládacích prvků je provedena pomocí událostmi řízené struktury. Vzhled tohoto panelu je na obrázku 43.



Obr. 43 Vzhled čelního panelu programu pro vzdálené ovládání osciloskopu

Poslední řídicí aplikace slouží k ovládání osciloskopu. Také v tomto případě je po vytvoření spojení mezi řídicí aplikací a simulátorem sesynchronizováno nastavení simulátoru s nastavením řídicí aplikace. Následně pokračuje program smyčkou, ve které je vytvořena událostmi řízení struktura. Tato struktura slouží k obsluze ovládacích prvků z čelního panelu. V této struktuře, je také událost „timeout“, při které dochází k vyčtení osciloskopem změřených hodnot a jejich zobrazení na obrazovce.



Obr. 44 Zdrojový kód programu pro vzdálené ovládání osciloskopu

7 Testování a ověření funkčnosti aplikace

Testování se provádí ze dvou důvodů. Tím prvním důvodem je ověření, že výsledný produkt splňuje předem stanovené vlastnosti a druhým důvodem je nalezení co největšího množství chyb. Pomocí testování můžeme pouze zjistit přítomnost chyby. Dalším krokem po testování je ladění kdy jsou jednotlivé chyby lokalizovány díky tomu je následně opravit, nebo zbránit jejich vzniku.

Prvními částmi, které byly otestovány subVI určené pro rozpoznání příkazů. U těchto bloků se testovalo, zda jsou rozpoznány všechny podporované příkazy, zda v případě chybné syntaxe je do zásobníku chyb uloží správná hodnota, popřípadě, zda je vyčtená ze zásobníku správná hodnota.

Dalšími samostatně testovanými částmi byly části obsahující samotnou simulaci funkcí přístrojů. Zde bylo testováno, zda funkce odpovídají specifikacím. Zda a tyto části fungují správně.

Testování celého systému bylo provedeno ve dvou režimech.

Prvním režimem byl offline mód simulátoru. Kdy bylo otestováno, zda jsou všechny ovládací prvky na čelních panelech správně funkční. V tomto režimu bylo také otestováno, zda vlastnosti jednotlivých přístrojů odpovídají specifikaci, která byla vytvořena na počátku.

Druhým režimem testování byl online mód simulátoru. V online režimu byla testována komunikace simulátorů s aplikací, která byla vytvořena jako vzorový příklad využití přístrojových ovladačů. Byla zde otestována funkčnosti jednotlivých přístrojových ovladačů. U těch bylo testováno, zda se správně vytvoří příkaz složením těla příkazu a převedené hodnoty do textového řetězce.

Pomocí druhého režimu, byla objevena nežádoucí vlastnost simulátoru. Touto vlastností bylo, zamrznutí programu v případě, že se uživatel pokusí simulátor vypnout v momentě kdy je k němu připojena aplikace pro vzdálené řízení. Tato vlastnost, byla odstraněna znemožněním uživateli vypnout simulátor v momentě, kdy je k němu připojena aplikace pro vzdálené řízení kteréhokoliv ze čtyř simulovaných přístrojů.

Závěr

Cílem této práce bylo nejen seznámení se s problematikou programování měřících přístrojů dle standardu SCPI a s problematikou ovládání měřících přístrojů přístrojovými ovladači standardu VISA Plug & Play, ale hlavně vytvoření programu, který může v rámci výuky nahradit svou funkcí skutečné přístroje, jejichž pořízení je finančně nákladné.

V úvodu práce jsou shrnuty výhody přístrojů, které jsou schopny komunikovat s počítačem. Následně jsou popsány možnosti při výběru fyzických rozhraní používaných pro komunikaci s měřicími přístroji. Jsou také vyjmenovány výhody a nevýhody jednotlivých rozhraní. Dalším zajímavým tématem teoretické části je vysvětlení problematiky SCPI příkazů. Jsou zde také popsány výhody používání standardu SCPI. Předposledním tématem, kterým se teoretická část zabývá je standard pro tvorbu přístrojových ovladačů Plug & Play. Tento standard popisuje vnitřní i vnější strukturu přístrojového ovladače. Dále rozděluje přístrojové ovladače do 6 skupin, dle jejich využití. Posledním tématem, kterým se práce v teoretické části zabývá, jsou vnitřní registry přístrojů, které informují o jeho stavu a vyskytlých událostech.

Praktická část této diplomové práce začíná vytipováním přístrojů vhodných pro vytvoření jejich simulace. Po vybrání čtyř přístrojů (zdroj napětí, generátor, multimetr a osciloskop) jsou navrženy a definovány jejich vlastnosti. U zdroje napětí bylo definováno minimální a maximální napětí, které tento zdroj může mít na výstupu. Dále u zdroje byl definován mód napěťové rampy. U generátoru byly vybrány typy průběhů, které bude umět generovat, rozpětí amplitudy signálu, frekvence, amplitudy šumu a velikost stejnosměrného offsetu. Jeden z požadavků na generátor bylo také to, aby uměl vygenerovat signál s měnící se frekvencí. Pro digitální multimetr byly vybrány veličiny, které tento multimetr bude schopen měřit u signálu, který bude přiveden. Dále byly specifikovány rozsahy multimetru. Vstup multimetru je volitelný a je možno vybrat mezi zdrojem napětí nebo generátorem. Posledním přístrojem je osciloskop, u toho přístroje byly definovány napěťové rozsahy a rozsahy časové konstanty. Dále byly blíže specifikovány funkce, které bude osciloskop podporovat. PO definování vlastností jednotlivých přístrojů byly vytvořeny příkazy pro jejich ovládání.

Po těchto přípravných krocích se práce zabývá vytvořením jednotlivých bloků programů simulující jednotlivé přístroje. Souběžně s tvorbou jednotlivých subVi pro simulaci přístrojů, byly vytvořeny části pro obsluhu komunikace. Ke komunikaci bylo vybráno rozhraní ethernet, protože jako jediné možná rozhraní nepotřebuje pro provoz komunikace další fyzické části a je možno u něj využít lokální komunikace. Každému z přístrojů byl přiřazen port pro komunikaci. Byly vybrány porty 9999,9998,9997,9996 a to protože tyto porty nejsou běžnými aplikacemi využívány, čímž se odstranila možnost kolize. Po vytvoření částí pro simulaci přístrojů, obsluhu komunikace a rozpoznání příkazů byly tyto jednotlivé části zkompletovány do jednoho funkčního bloku.

Další krokem bylo vytvoření přístrojových ovladačů, které budou sloužit k jednoduššímu naprogramování řídicí aplikace pro simulované přístroje. Bylo zapotřebí vytvoření hlavně ovladačů pro inicializaci spojení, pro jednotlivé příkazy nastavující vlastnosti přístrojů, pro příkazy, které vyčítají hodnoty z přístrojů, několika ovladačů s funkcemi pro obsluhu přístrojů a také ovladač pro ukončení spojení.

Po vytvoření simulátoru a přístrojových ovladačů, byly vytvořeny také příklady využití těchto ovladačů v podobě čtyř programů obsluhující jednotlivé přístroje.

Výsledné programy byly náležitě otestovány a to jak z hlediska splnění požadavků na jednotlivé přístroje tak i z hlediska samotné funkčnosti jednotlivých přístrojů. Byla také otestována komunikace simulovaných přístrojů s aplikacemi pro jejich ovládání.

Výsledná aplikace jistě může zastoupit fyzické přístroje v rámci výuky komunikace s měřicími přístroji.

V přílohách jsou seznamy příkazů s jejich správnou syntaxí a popisem k čemu slouží. V příloze je také uveden manuál pro obsluhu simulátoru.

Použité zdroje a literatura

- [1] ŽÍDEK, Jan. *Měřicí systémy a jejich programování*. Ostrava: VŠB-TUO, [200?]. Podpora komunikace s měřicími přístroji ve vývojovém prostředí LabView, s. 204-241.
- [2] *Zone.ni.com* [online]. 30.4.2008 [cit. 2010-01-10]. GPIB Hardware and Software Specifications. Dostupné z WWW: <<http://zone.ni.com/devzone/cda/tut/p/id/3388>>.
- [3] *Zone.ni.com* [online]. 8.2.2010 [cit. 2010-01-10]. Developing LabVIEW Plug and Play Instrument Drivers. Dostupné z WWW: <<http://zone.ni.com/devzone/cda/tut/p/id/3271>>.
- [4] PIEPER, John. *Automatic measurement control : A tutorial on SCPI and IEEE 488.2*. Mnichov : Rohde & Schwarz, [199?]. Status and error reporting, s. 158-184. ISBN 978-3-939837-02-2.

Seznam příloh

Příloha I	-	Příkazy pro ovládání zdroje napětí
Příloha II	-	Příkazy pro ovládání generátorů funkcí
Příloha III	-	Příkazy pro ovládání digitálního multimetru
Příloha IV	-	Příkazy pro ovládání osciloskopu
Příloha V	-	SCPI příkazy
Příloha VI	-	IEEE 488.2 příkazy
Příloha VII	-	Manuál pro obsluhu simulátoru

